An improved video watermarking algorithm with extraction using a mobile device camera

O.O. Evsutin [1], A.S. Melman [1], D.I. Podbolotov [1], A.G. Stankevich [1]
[1] National Research University Higher School of Economics, 101000, Moscow, Russia, Myasnitskaya Ulitsa, 20

Abstract

The use of screen-capture-resistant digital watermarks is a promising way to store information invisibly in a video stream for later retrieval by the user using a smartphone camera. However, the development of algorithms that implement this scenario is associated with the problem of balancing between the imperceptibility of embedding and robustness. A serious problem is the extraction of watermarks using a mobile device. Most people use the vertical positioning of the smartphone when shooting, which excludes the possibility of only marked video sequences entering the frame. The extraction algorithm first finds the screen area in the image and then extracts the watermark under various distortion conditions. This study proposes an approach to improve the efficiency of the algorithm for embedding digital watermarks into video data based on rectangular patterns, which provides resistance to screen shooting. The proposed approach to increasing the embedding imperceptibility provided an increase in the PSNR and SSIM values by 17.18 % and 7.90 %, respectively. The use of a neural network at the extraction stage reduced the BER value by 64.64 %.

Keywords: digital watermark, video, neural network, screen capture.

Citation: Evsutin OO, Melman AS, Podbolotov DI, Stankevich AG. An improved video watermarking algorithm with extraction using a mobile device camera. Computer Optics 2023; 47(6): 972-979. DOI: 10.18287/2412-6179-CO-1328.

Introduction

The rise in the number of devices connected to the network leads to a constant increase in the amount of generated digital data, including multimedia data. At the same time, the active development of immersive applications that combine the digital and physical worlds leads to the emergence of new ways of human interaction with digital content. The need to transition from the objects of the digital world to the objects of the real world and vice versa appears in various areas, such as industry, medicine, and education. Examples are augmented reality technologies for modeling the environment and information extraction from video streams and graphic objects in real-time.

Digital watermarking is a promising way to connect the physical and digital worlds. A watermark is some additional information that is embedded in a cover object [1, 2]. Various digital objects, such as images and video sequences, can be watermarked. Unlike quick-response (QR) codes, which have become widespread in recent years, digital watermarks usually do not make visible changes to the content and are stored in an invisible way.

The main indicators of the watermarking algorithm efficiency are imperceptibility and robustness. These indicators are mutually inverse, so the major task is to achieve a balance between them.

Watermarks can be used both to protect the rights to a digital object [3, 4] and to store additional information, for example, links to product information in promotional materials [5, 6]. If the watermark is used to imperceptibly store additional information in a video stream, the expected user's scenario for obtaining such information is as follows:

- The user points his or her smartphone camera at the screen of the device showing the watermarked video;
- A certain number of video frames are being captured;
- Some software analyzes the captured frames, extracts the watermark, and provides the user with the resulting information.

This scenario is illustrated in Fig. 1. It imposes additional requirements on the robustness of the watermarking algorithm since the watermark must be resistant to the capture from the screen (so-called "screen-cam" or "screen-shooting" attack). Such shooting distorts the color rendition of the captured frame and is accompanied by the image rotation to an arbitrary angle as well as the frame size changes caused by different distances from the smartphone camera to the display.

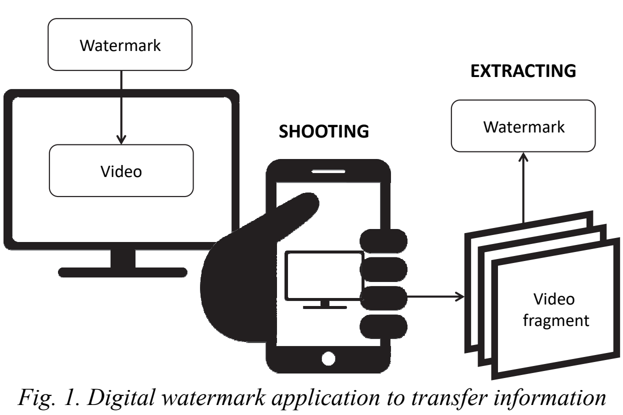

Fig. 1. Digital watermark application to transfer information in a video

In real life, the user is unlikely to spend a lot of time looking for a camera position in which only the correct video stream enters the lens. At the same time, the smartphone is located vertically in most cases, which eliminates the possibility of such a positioning of the device, in which unnecessary objects do not enter the picture. Therefore, an important task is to determine the screen area showing the video in the picture from the mobile device camera. The effectiveness of algorithms that implement the search for the screen location in the image depends on many factors, including the contrast between the screen and the background, the rotation angle, the presence of other objects in the image, etc.

In [7], an algorithm for embedding watermarks into video data based on rectangular patterns was presented. It provides an increased resistance to screen-shoot attack. In our study, we propose a modification of this algorithm. The proposed modification has the following features:

- The distortion level of the marked video is reduced due to the choice of the color component for embedding and applying a blur to the watermark;
- The accuracy of determining the area of the screen showing the watermarked video in the picture is increased through the use of a neural network.

The rest of the paper is organized as follows. In Section 1, we present a brief overview of the research in the field of robust video watermarking. In Section 2, we describe the previous algorithm [7]. Section 3 presents the proposed approaches to improve the algorithm. Section 4 presents the experimental results and their discussion. The conclusion summarizes the study.

1. Related work

There are many algorithms for embedding digital watermarks into digital images and videos. One of the important characteristics of watermarking algorithms is the ability to extract data without using the original cover object or watermark. Such algorithms are called blind. If the initial data is needed for extraction, then such algorithms are called non-blind. The requirements for watermarking algorithms vary by application. Most watermark applications related to data protection assume non-blind extraction, since they involve comparing the extracted watermark with some reference value [8, 9]. However, the scenario considered in this study does not require any additional information from the user, so only blind watermarking algorithms are discussed below.

Robustness is one of the key performance indicators of watermarking algorithms. Authors of watermarking schemes for images and video sequences use many different approaches to improve the robustness of embedding, including cellular automata [10], edge detection methods [11], metaheuristic optimization methods [12 – 14], and neural networks [15 – 17].

In recent years, there has been an increase in the interest of researchers in the topic of watermarking algorithms that are resistant to digital-to-analog and analog-to-digital conversions, as well as conversions associated with the transition from one digital representation to another caused by shooting from the screen. Such algorithms designed for working with images can be divided into screen-cam or screen-shooting robust [18, 19] and print-cam or print-scan robust [20, 21].

Video is a set of individual images (frames), so watermarking algorithms designed for images can also be used to embed watermarks into video sequences. However, the special properties of video data motivate researchers to develop algorithms that work only with this type of data.

In [22], an example of a screen-cam robust watermarking algorithm for video is presented. The authors propose to embed watermarks into B-frames of MPEG-2 video. The average value of the low-frequency discrete cosine transform coefficients of video is modulated according to the bits of the watermark.

The authors of [23] propose a watermarking algorithm for high-definition videos. They use the adaptive embedding strength according to Watson's enhanced visual model to increase the imperceptibility and robustness of watermarked videos. In the watermark-detecting scheme, temporal synchronization and adaptive detecting areas are used.

Paper [24] proposes a watermarking scheme that is resistant to frame blending and projection attacks. The authors of the study discuss in detail the distortions caused by capturing video from a camera and propose an approach based on video segmentation using geometrically invariant functions.

In [25], a robust watermarking scheme is proposed to combat piracy. The watermark is formed taking into account the date and time of the screening, the cinema's registration number, and the six-digit zonal improvement code of the cinema location. The embedding algorithm is based on changing the coefficients of the discrete wavelet transform.

The authors of [26] propose an approach that uses the integration of the dual-tree complex wavelet transform and singular value decomposition to achieve robustness against geometric attacks. The algorithm is resistant to timing attacks, such as frame rate conversion and video shooting.

The examples of video watermarking algorithms discussed above demonstrate high performance and successfully resist the attack of video recording on the camera. However, the authors do not consider the aspects of determining the screen area on the frame, since it is assumed that only the target video stream enters the camera lens. When shooting video using a smartphone, it is not possible to realize such ideal conditions. Therefore, in this study, we focus on solving the problem of determining the area of the screen that demonstrates the marked video.

2. Previous algorithm

In [7], an algorithm for embedding digital watermarks into video data is proposed which ensures resistance to shooting with a smartphone camera. In this section, we discuss the main ideas of this algorithm.

The authors of [7] embed a binary watermark into the spatial domain of the frame by applying a pattern consisting of rectangles whose brightness values encode the transmitted information. Each rectangle encodes one bit of data. The pattern size corresponds to the video frame size. The angular positions of the pattern are not used to encode information and correspond to fixed values. The pattern corresponding to the bits of the watermark is called positive, and its inverted version is called negative. An example of a watermark and its corresponding positive and negative patterns is shown in Fig. 2. Note that the rectangles containing 0 and 1 in Fig. 2 correspond not to individual pixels, but to blocks of pixels in the original frame. The size and number of blocks are determined by the frame size and embedding capacity.

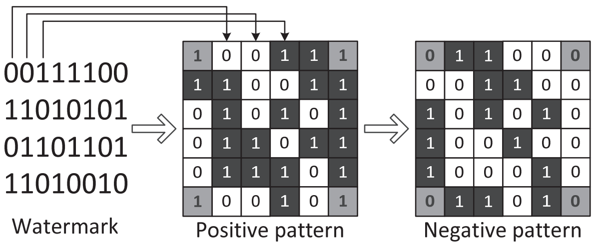

Fig. 2. Embedding pattern

The data is embedded into the blue channel of the RGB color model. When embedding a watermark, the patterns are sequentially added to the frames of the original video, alternating as follows: positive, negative, negative, and positive. Thus, the first frame corresponds to a positive pattern, the second frame corresponds to a negative one, the third corresponds to a negative one, the fourth corresponds to a positive pattern, and then the sequence repeats (the fifth is positive, the sixth is negative, etc.). The pattern values are preliminarily multiplied by the value T_i, which is calculated by the following formula:

$$T_i = \begin{cases} \mathrm{round}\left(\dfrac{S-d_i}{2}\right), \text{if } c_i = 1 \text{ and } d_i < S; \\ 0, \text{if } c_i = 1 \text{ and } d_i \geq S; \\ \mathrm{round}\left(\dfrac{S+d_i}{2}\right), \text{if } c_i = 0 \text{ and } d_i > -S; \\ 0, \text{if } c_i = 0 \text{ and } d_i \leq -S; \end{cases} \qquad (1)$$

where S is an embedding strength, d_i is a mean value of pixels in the i-th block in the difference frame of successive two frames in one unit for embedding, c_i is a watermark bit.

The watermark extraction process consists of two steps: detecting the screen area in the smartphone camera image and extracting embedded bits. The input of the extraction procedure is M frames taken by the mobile device camera. Next, differential frames are computed for $M-1$ pairs of adjacent frames. The authors use an approach based on the search for frame blocks containing certain features of embedded patterns, the use of morphological operations and rotation operations to determine the screen area. Extra information is removed from the image and only frames of marked video after preprocessing are used for extraction.

The differential frames with the highest average pixel intensity are divided into $X \times Y$ blocks. This procedure divides the differential frame into regions, each of which contains one bit of the watermark. Watermark bits c_i are extracted from blocks of differential frames, excluding corner blocks, by the formula

$$c_i = \begin{cases} 1, \text{if } b_i > 0 \text{ and } mc > 0; \\ 0, \text{if } b_i < 0 \text{ and } mc > 0; \\ 1, \text{if } b_i > 0 \text{ and } mc < 0; \\ 0, \text{if } b_i < 0 \text{ and } mc < 0; \end{cases} \qquad (2)$$

where mc is a mean value of pixel values of corner blocks, and b_i is a mean value of pixel values of corresponding block.

3. The proposed approach

In this section, we consider the algorithm [7] and propose a number of modifications.

3.1. An approach to reduce the level of frame distortion

The authors of the algorithm [7] use the blue component of the RGB model to embed a watermark. Some videos or their individual frames are characterized by the predominance of some color, for example, due to the use of color filters. The predominance of any color on video frames leads to the appearance of embedding artifacts that are visible to the naked eye when one color component is changed. This is shown in Fig. 3-4. In Fig. 3, embedding the watermark into the green and blue components results in noticeable distortion. In Fig. 4, distortion occurs when the watermark is embedded in the green RGB component. Similar distortions occur for frames of other videos.

Fig. 5 presents the results of color palette cluster analysis of the frames from Fig. 3*a* and Fig. 4*a* using the k-means method. We use ImageColorAnalysisTool for Matlab [27], and select only one “averaged” color in the image. The algorithm selects points in the image whose color is closest to the current point and calculates the “averaged” color of these points. Processing continues until the distances between the averaged points are optimally distributed. The empirical study shows that frame distortion is least noticeable when the watermarked color component contributes neither the most nor the least to the final “averaged” color.

In the image palette in Fig. 3*a*, the green component contributes the most, and the blue component contributes the least. Therefore, the recommended component for embedding the watermark is the red one. In the image palette in Fig. 4a, the red component contributes the most, and the green component contributes the least. Therefore, it is recommended to embed the watermark into the blue component.

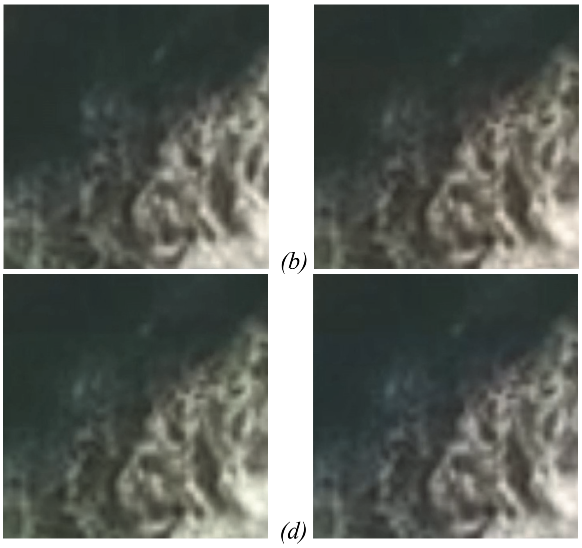
(a) (b) (c) (d)

Fig. 3. Frame fragments of video 1: a) before embedding; b) watermarked R-component; c) watermarked G-component; d) watermarked B-component

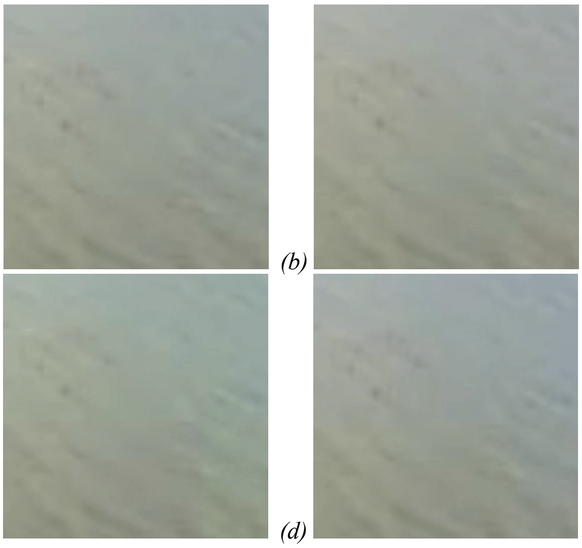
(a) (b) (c) (d)

Fig. 4. Frame fragments of video 2: a) before embedding; b) watermarked R-component; c) watermarked G-component; d) watermarked B-component

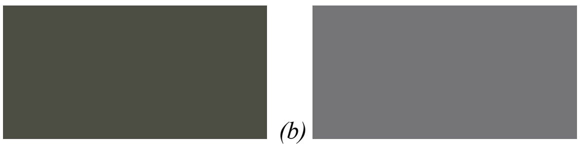
(a) (b)

Fig. 5. The result of cluster analysis: a) RGB(71,76,59); b) RGB(116,114,115)

Note: if the video consists of a sequence of frames whose palettes differ dramatically, the blue component remains the most suitable for embedding.

As can be seen, for example, in Fig. 3*c* and 4*c*, the use of rectangular patterns for embedding data can lead to frame distortions that look like rectangular areas with a changed color tone. A significant change in the intensity of one color component of an image or frame is quite noticeable to the human eye. We apply one of the image blur techniques, Gaussian blur, to the pattern to solve this problem. Fig. 6 illustrates an example of a pattern before and after blurring at $\sigma = 20$, where σ is a standard deviation of Gaussian distribution. Fig. 7 illustrates the effect of applying a Gaussian filter with $\sigma = 20$ to watermarked frames from Fig. 3*c* and 4*c*. Applying blur significantly smoothed out the edges of the square areas.

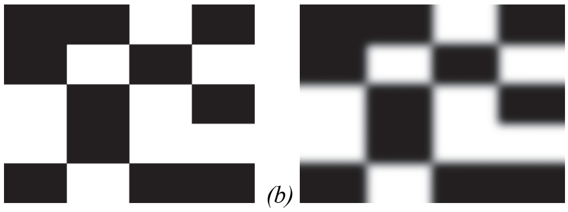
(a) (b)

Fig. 6. Example of a pattern: a) without blur; b) with blur

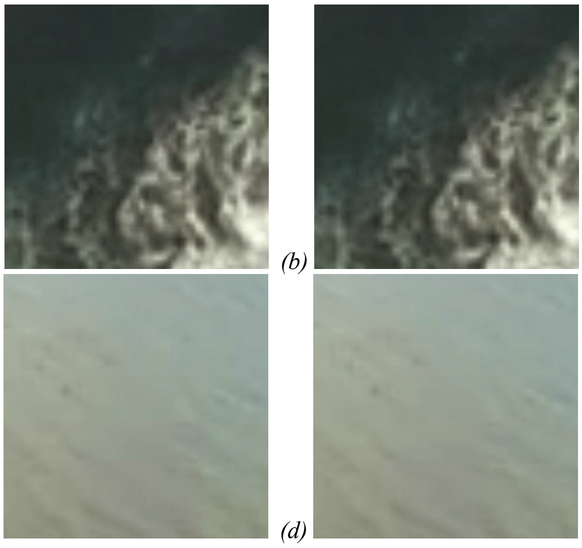
(a) (b) (c) (d)

Fig. 7. Applying Gaussian blur to the pattern: a) watermarked frame of video 1 without blur; b) watermarked frame of video 1 with blur; c) watermarked frame of video 2 without blur; d) watermarked frame of video 2 with blur

3.2. An approach to improving the accuracy of determining the screen area in a picture

The authors of the algorithm [7] assessed its effectiveness when shooting from a tripod. These are ideal conditions that cannot be achieved in practice. Therefore, to increase the robustness of embedding, it is necessary to find the area of the watermarked frame in the image under conditions close to real ones.

In recent years, neural networks have become a popular tool for solving various image processing problems [28, 29]. U-net architecture neural network [30] is applied to improve the accuracy of screen area detection. This architecture was introduced in 2015 for object segmentation in an image. It provides high-speed and achieves good results even when using a small amount of data. In this study, it is important to use a neural network that segments objects instead of just finding them. This is necessary to determine the area of the screen when it is rotated and does not look like a rectangle.

The architecture of this network can be divided into two parts: a contracting path and an expansive path. The first one is a classic convolutional neural network, it consists of successive convolutions with a kernel $(3,3)$ and ReLU activation and a subsequent MaxPooling to reduce the resolution. The second one consists of a series of transposed convolutional layers with a kernel $(2,2)$, a concatenation with the corresponding tensor from the first part of the network, and two subsequent convolutions with a kernel $(3,3)$ and ReLU activation. The output layer for the segmentation problem is convolution with kernel $(1,1)$ and SoftMax activation.

For the purposes of this study, the classic U-net architecture has been modified as follows:

- The input image resolution was set to 240×160 pixels;
- The number of features at each step was reduced from 128, 256, 512, and 1024 to 16, 32, 48, and 64 respectively.

The specified resolution is one of the standard presets of the official camera plugin in the flutter mobile UI development kit. Thus, there is no need to resize the image after receiving it from the camera before passing it to the input of the neural network. This architecture supports processing of 43 frames per minute on average when running on a mobile device on 8 threads.

We formed a new dataset for training and testing a neural network. To prepare the dataset, 29 videos of different lengths were recorded. In all of them, the screen of the device was filled with green color. Each video was split into images at a frequency of 2 frames per second. The green area of the screen was replaced with one of 100 unique images. Fig. 8 illustrates the markup process.

Fig. 8. Frame markup example

4. Experimental results

For the experiments, 6 different videos from open sources were used. They were cropped with a video editor [31]. The displayed and captured video frame rate was 30 fps. The resolution of the captured video was set to 240×320, which is the minimum resolution available for Android devices. This choice of resolution allows us to reduce the processing time, since no significant resolution transformation is required for the input of the neural network. In accordance with the selected resolution, the size of one pattern block was 48×80 for a watermark sequence of 16 bits and 48×64 for a sequence of 21 bits.

Watermark embedding and watermarked video demonstration were performed on a laptop with an Intel i5, 2.10 GHz processor and 16 GB RAM (OS Windows). For extraction, a smartphone with a HiSilicon Kirin 970 processor and 4 GB RAM (OS Android) was used. The shooting was carried out without using a tripod, the smartphone camera was located at a distance of about 50 cm from the screen of laptop, and the smartphone was placed parallel to the screen.

Standard metrics such as peak signal-to-noise ratio (PSNR) and index of structural similarity (SSIM) were used to assess the imperceptibility of the embedding. The PSNR value is calculated as follows:

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right), \tag{3}$$

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{i=1}^{M \times N}\left(C_i - S_i\right)^2, \tag{4}$$

where $M \times N$ are width and height of a frame, C_i is a pixel value of the original frame, and S_i is a pixel value of the marked frame.

The SSIM value is calculated using the formula

$$\mathrm{SSIM} = \frac{\left(2\mu_C\mu_S + K_1\right)\times\left(2\sigma_{CS} + K_2\right)}{\left(\mu_C^2 + \mu_S^2 + K_1\right)\times\left(\sigma_C^2 + \sigma_S^2 + K_2\right)}, \tag{5}$$

where μ_C is a mean pixels value of an original frame, μ_S is a mean pixels value of a marked frame, σ_C^2 is a variance of pixel values the original frame, σ_S^2 is a variance of pixel values of the marked frame, σ_{CS} is a covariance of both frames, K_1 and K_2 are constants.

We used the Bit Error Rate (BER) metric to assess the robustness of embedding. The BER value shows the ratio of the number of incorrectly extracted bits to the total size of a watermark and is calculated as follows:

$$\mathrm{BER} = \frac{B_e}{B}, \tag{6}$$

where B is a watermark size, and B_e is a number of extraction errors.

We estimated the dependence of the original algorithm performance [7] on the embedding strength S before applying the proposed modifications. The PSNR and SSIM values are presented in Table 1 for watermark sizes of 16 and 21 bits, respectively. Experiments showed that the original algorithm provides higher values of imperceptibility metrics at lower embedding strengths.

Tab. 1. PSNR and SSIM values for different S values

S	PSNR, dB	SSIM
16 bits		
1	44.9017	0.9823
3	40.6568	0.9703
6	36.2966	0.9372
9	33.2896	0.8902
21 bits		
1	43.9662	0.9803
3	40.1012	0.9679
6	35.9905	0.9343
9	33.0861	0.8868

Fig. 9 shows BER values versus Gaussian blur σ value for different embedding strengths. As follows from the graph, until the σ value reaches 30, no errors occur when extracting the watermark.

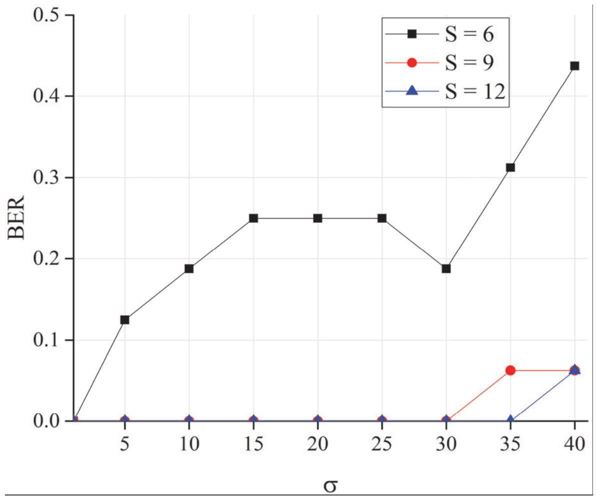

Fig. 9. BER versus Gaussian blur σ value

Blurring can significantly reduce the distortion level of watermarked frames. Thus, when embedding a digital watermark with a strength $S=9$ and $\sigma=30$, it is possible to achieve a distortion level close to $S=3$ without blurring. Tab. 2 contains the PSNR and SSIM values for the embedding strength value $S=9$ and a 16-bit watermark. The increase in the values of the PSNR and SSIM metrics was 17.18 % and 7.90 %, respectively.

Tab. 2. PSNR and SSIM values for different σ values

σ	PSNR, dB	SSIM
1	36.75	0.9441
5	37.09	0.9474
10	37.52	0.9508
15	37.97	0.9541
20	38.39	0.9567
25	38.73	0.9588
30	39.01	0.9605

We also performed the following experiment to evaluate the effectiveness of the proposed approach in improving the accuracy of watermark extraction. We embedded the same watermarks without blurring in the B-component of different videos and tried to extract them using the original algorithm and our modification. We did not use a tripod for shooting to ensure conditions were as close to real as possible. Fig. 10 shows the result of the experiment. As can be seen in the figure, the original algorithm demonstrates a higher BER when shooting without a tripod. The modified version of the algorithm provides a BER value of less than 0.1 at the embedding strength $S=6$, corresponding to acceptable embedding imperceptibility. On average, the BER value decreased by 64.64 %.

We also evaluated the effectiveness of extracting a watermark from a video obtained using a new approach to improve the embedding imperceptibility. We used the most appropriate RGB component for embedding, parameter $S=9$, and Gaussian blur parameter $\sigma=30$. We showed above that these embedding parameter values provide a high imperceptibility level. The average BER value for the original algorithm is 0.3958, and it is 0.2101 for the modified version. Thus, the proposed approach makes it possible to simultaneously achieve high invisibility of embedding and a significant increase in robustness compared to the original algorithm.

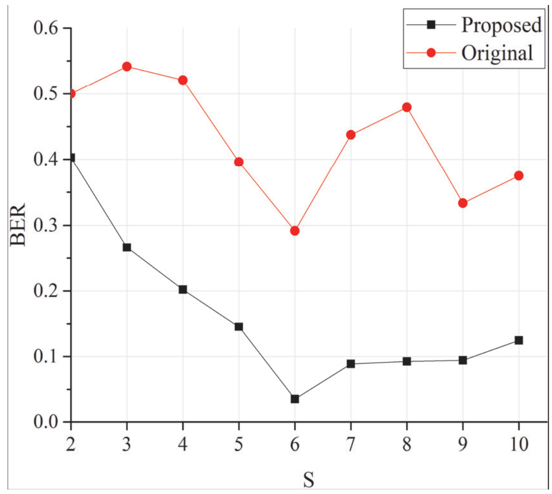

Fig. 10. BER versus embedding strength S

It is important to note that previous studies used a tripod to perform similar experiments, which is not consistent with the basic scenario we described for extracting watermarks from videos. In our study, we did not use a tripod, and the shooting was done manually, which creates additional frame distortions. Experiments showed that under similar conditions, our modification is better able to resist such distortions.

Thus, we significantly improved the efficiency of the original algorithm [7]. However, it should be noted that the small capacity of the algorithm remains an unsolved problem. Each rectangular block in the pattern contains only 1 bit of information, and the capacity is directly related to the number of blocks. An increase in the number of blocks leads to a decrease in robustness. A promising way to increase the embedding capacity is to increase the hiding space. In this case, the target binary sequence is split into separate fragments and sequentially embedded in every M video frames. At the extraction stage, individual fragments are combined into one watermark sequence. Increasing the resolution of captured frames will increase the accuracy of watermark extraction. However, in this case, it is required to develop a new lightweight neural network architecture in order to ensure high performance of the scheme on various mobile devices. The problem of increasing the embedding capacity will be considered in future studies.

Conclusion

In this study, the modification of the video watermarking algorithm based on rectangular patterns, which is resistant to screen shooting, was proposed [7]. The proposed approach to reduce the level of watermarked video distortion by choosing an appropriate color component and Gaussian blurring of the watermark improved the imperceptibility of data embedding. The approach to improve the accuracy of determining the screen area in the picture through the use of a neural network increased the reliability of the embedded data transmission. Thus, the proposed modification has increased the practical applicability of the algorithm for the main use case. In the future, it is planned to increase the capacity and robustness of the algorithm. In particular, it is planned to explore different ways to increase the hiding space and consider different families of patterns, as well as expand the use of neural networks.

Acknowledgments

This work was supported by the Russian Science Foundation, project no. 21-71-10113.

References

[1] Egorova AA, Fedoseev VA. A classification of semi-fragile watermarking systems for JPEG images. Computer Optics 2019; 43(3): 419-433. DOI: 10.18287/2412-6179-2019-43-3-419-433.

[2] Vybornova YD, Sergeev VV. A new watermarking method for vector map data. Computer Optics 2017; 41(6): 913-919. DOI: 10.18287/2412-6179-2017-41-6-913-919.

[3] Melman A, Evsutin O, Shelupanov A. An authorship protection technology for electronic documents based on image watermarking. Technologies 2020; 8(4): 79. DOI: 10.3390/technologies8040079.

[4] Mitekin VA. An algorithm for generating digital watermarks robust against brute-force attacks. Computer Optics 2015; 39(5): 808-817. DOI: 10.18287/0134-2452-2015-39-5-808-817.

[5] Choi Y-H, Kim D, Rho S, Hwang E. Converting image to a gateway to an information portal for digital signage. Multimed Tools Appl 2014; 71(1): 263-278. DOI: 10.1007/s11042-012-1315-6.

[6] Chen Y-Y, Chi K-Y, Hua K-L. Design of image barcodes for future mobile advertising. EURASIP J Image Video Process 2017; 2017: 11. DOI: 10.1186/s13640-016-0158-x.

[7] Iwata M, Mizushima N, Kise K. Practical watermarking method estimating watermarked region from recaptured videos on smartphone. IEICE Trans Inf Syst 2017; E100.D(1): 24-32. DOI: 10.1587/transinf.2016MUP0008.

[8] Sridhar B, Syambabu V. Security enhancement in video based on gatefold technique for copyright protection. Multimed Tools Appl 2021; 80(14): 8241-8256. DOI: 10.1007/s11042-020-09909-z.

[9] Pandey MK, Parmar G, Gupta R, Sikander A. Non-blind Arnold scrambled hybrid image watermarking in YCbCr color space. Microsyst Technol 2019; 25(8): 3071-3081. DOI: 10.1007/s00542-018-4162-1.

[10] Priya C, Ramya C. Robust and secure video watermarking based on cellular automata and singular value decomposition for copyright protection. Circuits Syst Signal Process 2021; 40(5): 2464-2493. DOI: 10.1007/s00034-020-01585-6.

[11] Gong L-H, Tian C, Zou W-P, Zhou N-R. Robust and imperceptible watermarking scheme based on Canny edge detection and SVD in the contourlet domain. Multimed Tools Appl 2021; 80(1): 439-461. DOI: 10.1007/s11042-020-09677-w.

[12] Kang X, Chen Y, Zhao F, Lin G. Multi-dimensional particle swarm optimization for robust blind image watermarking using intertwining logistic map and hybrid domain. Soft Comput 2020; 24(12): 10561-10584. DOI: 10.1007/s00500-019-04563-6.

[13] Cedillo-Hernandez M, Cedillo-Hernandez A, Garcia-Ugalde FJ. Improving DFT-based image watermarking using particle swarm optimization algorithm. Mathematics 2021; 9(15): 1795. DOI: 10.3390/math9151795.

[14] Singh P, Devi KJ, Thakkar HK, Santamaría J. Blind and secured adaptive digital image watermarking approach for high imperceptibility and robustness. Entropy 2021; 23(12): 1650. DOI: 10.3390/e23121650.

[15] Zhong X, Huang P-C, Mastorakis S, Shih FY. An automated and robust image watermarking scheme based on deep neural networks. IEEE Trans Multimedia 2021; 23: 1951-1961. DOI: 10.1109/TMM.2020.3006415.

[16] Zhu L, Wen X, Mo L, Ma J, Wang D. Robust location-secured high-definition image watermarking based on key-point detection and deep learning. Optik 2021; 248: 168194. DOI: 10.1016/j.ijleo.2021.168194.

[17] Sinhal R, Jain DK, Ansari IA. Machine learning based blind color image watermarking scheme for copyright protection. Pattern Recogn Lett 2021; 145: 171-177. DOI: 10.1016/j.patrec.2021.02.011.

[18] Chen W, Zhu C, Ren N, Seppänen T, Keskinarkaus A. Screen-cam robust and blind watermarking for tile satellite images. IEEE Access 2020; 8: 125274-125294. DOI: 10.1109/ACCESS.2020.3007689.

[19] Fang H, Zhang W, Zhou H, Cui H, Yu N. Screen-shooting resilient watermarking. IEEE Trans Inf Forensics Secur 2019; 14(6): 1403-1418. DOI: 10.1109/TIFS.2018.2878541.

[20] Pramila A, Keskinarkaus A, Takala V, Seppänen T. Extracting watermarks from printouts captured with wide angles using computational photography. Multimed Tools Appl 2017; 76(15): 16063-16084. DOI: 10.1007/s11042-016-3895-z.

[21] Huang K, Tian X, Yu H, Yu M, Yin A. A high capacity watermarking technique for the printed document. Electronics 2019; 8(12): 1403. DOI: 10.3390/electronics8121403.

[22] Choi D, Do H, Choi H, Kim T. A blind MPEG-2 video watermarking robust to camcorder recording. Signal Process 2010; 90(4): 1327-1332. DOI: 10.1016/j.sigpro.2009.10.009.

[23] Li L, Dong Z, Lu J, et al. AN H.264/AVC HDTV watermarking algorithm robust to camcorder recording. J Vis Commun Image Represent 2015; 26: 1-8. DOI: 10.1016/j.jvcir.2014.08.009.

[24] Gaj S, Rathore AK, Sur A, Bora PK. A robust watermarking scheme against frame blending and projection attacks. Multimed Tools Appl 2017; 76(20): 20755-20779. DOI: 10.1007/s11042-016-3961-6.

[25] Dubey N, Modi H. A robust discrete wavelet transform based adaptive watermarking scheme in YCbCr color space against camcorder recording in Cinema/Movie Theatres. Eng Sci 2021; 15: 116-128. DOI: 10.30919/es8d491.

[26] Asikuzzaman Md, Mareen H, Moustafa N, Choo K-KR, Pickering MR. Blind camcording-resistant video watermarking in the DTCWT and SVD domain. IEEE Access 2022; 10: 15681-15698. DOI: 10.1109/ACCESS.2022.3146723.

[27] Dorad. Image Color Analysis Tool. MATLAB Central File Exchange. 2022. Source:

<https://www.mathworks.com/matlabcentral/fileexchange/75116-image-color-analysis-tool>.

[28] Kalinina MO, Nikolaev PL. Book spine recognition with the use of deep neural networks. Computer Optics 2020; 44(6): 968-977. DOI: 10.18287/2412-6179-CO-731.

[29] Makarkin M, Bratashov D. State-of-the-art approaches for image deconvolution problems, including modern deep learning architectures. Micromachines (Basel) 2021; 12(12): 1558. DOI: 10.3390/mi12121558.

[30] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In Book: Navab N, Hornegger J, Wells WM, Frangi AF, eds. Medical image computing and computer-assisted intervention – MICCAI 2015. Cham: Springer International Publishing; 2015: 234-241. DOI: 10.1007/978-3-319-24574-4_28.

[31] Video tools. Online [In Russian]. Source: <https://clideo.com/>.

Authors' information

Oleg Olegovich Evsutin (b. 1987) graduated from the Tomsk State University of Control Systems and Radioelectronics in 2009 and received his Candidate in Engineering degree from the Tomsk State University in 2012. He works as the head of Information Security of Cyber-Physical Systems department, National Research University Higher School of Economics (HSE), Moscow, Russia. Research interests: cybersecurity, digital image processing, digital watermarking, digital steganography. E-mail: *oevsyutin@hse.ru* .

Anna Sergeevna Melman (b. 1994) graduated from the Tomsk State University of Control Systems and Radioelectronics in 2018. She works as a junior researcher of Information Security of Cyber-Physical Systems department, HSE, Moscow, Russia. Research interests: digital image processing, digital watermarking, digital steganography. E-mail: *amelman@hse.ru* .

Dmitriy Igorevich Podbolotov (b. 1998) is a 4th year undergraduate student in the Computer Science and Engineering program, HSE, Moscow, Russia. His research interests include digital watermarking, neural networks.
E-mail: *dipodbolotov@edu.hse.ru* .

Alexey Gennadievich Stankevich (b. 2000) graduated from HSE in 2022 with a bachelor's degree in Applied Mathematics. His research interests include digital watermarking, neural networks. E-mail: *agstankevich@edu.hse.ru* .

Received April 27, 2023. The final version – July 7, 2023.