Detection of deleted frames on videos using a 3D convolutional neural network

V. Voronin, R. Sizyakin, A. Zelensky, A. Nadykto, I. Svirin

V. Voronin, R. Sizyakin, A. Zelensky, A. Nadykto, I. Svirin, "Detection of deleted frames on videos using a 3D convolutional neural network," Proc. SPIE 10802, Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies II, 108020U (2 November 2018); doi: 10.1117/12.2326806

SPIE.

Event: SPIE Security + Defence, 2018, Berlin, Germany

Detection of deleted frames on videos using a 3D Convolutional Neural Network

V. Voronin[a], R. Sizyakin[a], Zelensky[b], A. Nadykto[b], I. Svirin[c]
[a]Lab. «Mathematical methods of image processing and intelligent computer vision systems»,
Don State Technical University, Rostov-on-Don, Russian Federation
[b]Moscow State University of Technology "STANKIN", Moscow, Russia
[c]CJSC Nordavind, Moscov, Russian Federation

ABSTRACT

Digital video forgery or manipulation is a modification of the digital video for fabrication, which includes frame sequence manipulations such as deleting, insertion and swapping. In this paper, we focus on the detection problem of deleted frames in videos. Frame dropping is a type of video manipulation where consecutive frames are deleted to skip content from the original video. The automatic detection of deleted frames is a challenging task in digital video forensics. This paper describes an approach using the spatial-temporal procedure based on the statistical analysis and the convolutional neural network. We calculate the set of different statistical rules for all frames as confidence scores. Also, the convolutional neural network used to obtain the output scores. The position of deleted frames determines based on the two score curves for per frame clip. Experimental results demonstrate the effectiveness of the proposed approach on a test video database.

Keywords: forgery detection, CNN, video manipulation.

1. INTRODUCTION

Currently, with the rapid development of mobile and portable video capture technology, the amount of video material obtained with it is growing. One of the conditions for these videos is their authenticity. Digital video forgery or manipulation is a modification of the digital video for fabrication, which includes frame sequence manipulations such as deleting, insertion and swapping [1,2]. Frame dropping is a type of video manipulation where consecutive frames are deleted to skip content from the original video. The automatic detection of deleted frames is a challenging task in digital video forensics.

Most common temporal tampering in videos (Fig.1):

- Frame dropping or frame removal;
- Frame swapping;
- Frame copy or frame addition;
- Frame replacement.

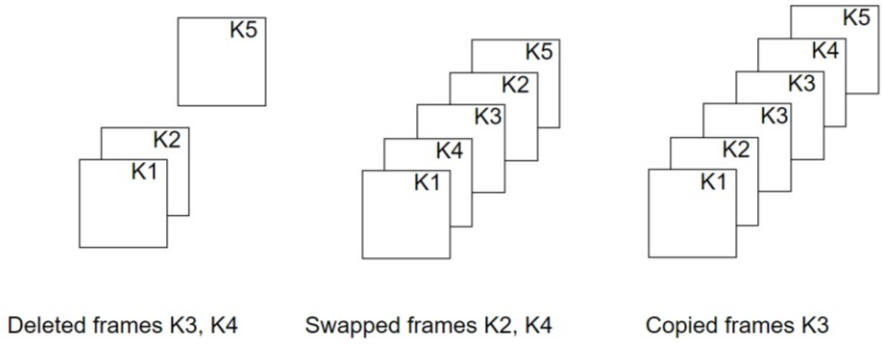

Figure 1. Temporal tampering in videos.

Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies II, edited by Henri Bouma, Radhakrishna Prabhu, Robert James Stokes, Yitzhak Yitzhaky, Proc. of SPIE Vol. 10802, 108020U

Since there are a lot of ways to manipulate video data, in this paper we consider only the problem of automatic frame detection, with the estimated time gap. This gap is characterized by a sharp change of spatial information, as well as the loss of correlation between adjacent frames.

There are several basic schemes for detection frames removal [1-8]: watermarking-based, learning-based, threshold-based, hashing-based.

One of the first works in this field is the work [9]. In this paper, the authors, based on the inter-frame difference of brightness histograms, find a gap in the correlation component, which could indicate the location of the proposed gluing. Since the user sets the threshold value in a heuristic way, the method requires a large amount of test data to select the optimal value, which is not always achievable in practice. The work [10] is a modification of the work [9], which consists of the subsequent processing of the obtained result to reduce false alarms. The threshold value is selected by the method proposed in [11]. After the places of the proposed gluing are found, four conditions for the spatial blocks into which adjacent frames are divided are checked. The first assumption is that the detection was due to the rapid movement of objects in the frame, or gluing. The second is the assumption that the background is homogeneous and stationary and does not contain glues. The third is that the background is movable and also does not contain glues. The last assumption is that the texture elements in the frame are also stationary and do not contain modifications. In this paper, the authors partially got rid of the dependence of setting the threshold value for the preliminary localization of glues. However, the user's participation is necessary to set the threshold for checking the above assumptions. Also, it should be noted that based only on the assessment of the brightness histogram; it is not always possible to achieve the desired result. This is because a sharp change in brightness often leads to false alarms. In [12], the authors use a modification of the texture operator LBP [13], as well as inter-frame correlation to localize the glues in the video sequence. The texture operator LBP allows giving robustness to the method of lighting differences in the frame [14]. Thus, the original LBP descriptor [15] is calculated by comparing each pixel with the Central one, which is taken as a threshold value, in a local area of 3 by 3 pixels. If the center pixel is less than or equal to the neighbor pixel, it is set to 1, otherwise 0. The modification is to increase the radius of the pixels, which are compared with the Central pixel, as well as the use of only nine templates, which carry the most informative about the texture features of the image and can reduce the number of non-informative bins.

One of the main drawbacks of the described methods is that they use only one characteristic as a base, and on which they later rely to detect time gaps in the video sequence.

The objective of our work is to develop a new approach for the detection of deleted frames on videos using a set of statistical characteristics and the convolutional neural network.

The rest of the paper is organized as follows. The proposed action recognition method is described in section 2. Section 3 presents some experimental results and conclusions are given in section 4.

2. PROPOSED METHOD

This paper describes a framework for digital video forgery or manipulation (see Fig. 2). We propose an approach for the detection of deleted frames in videos. The proposed algorithm is a two-stage procedure: (a) spatial-temporal analysis based on the statistical analysis and (b) the convolutional neural network for frame drop detection.

The algorithm of the described method is presented in Figure 3. There are several basic steps. At the training step, the CNN takes 9-frame video clips from the dataset, and produces two outputs, “frame deleted” or “no frame deleted”. At the testing step, we calculate the set of different statistical rules for all frames as confidence scores. Also, the learned convolutional neural network used to obtain the output scores. Based on the two score curves, we calculation multiplication between them and use threshold for detection deleted frames for per frame clip.

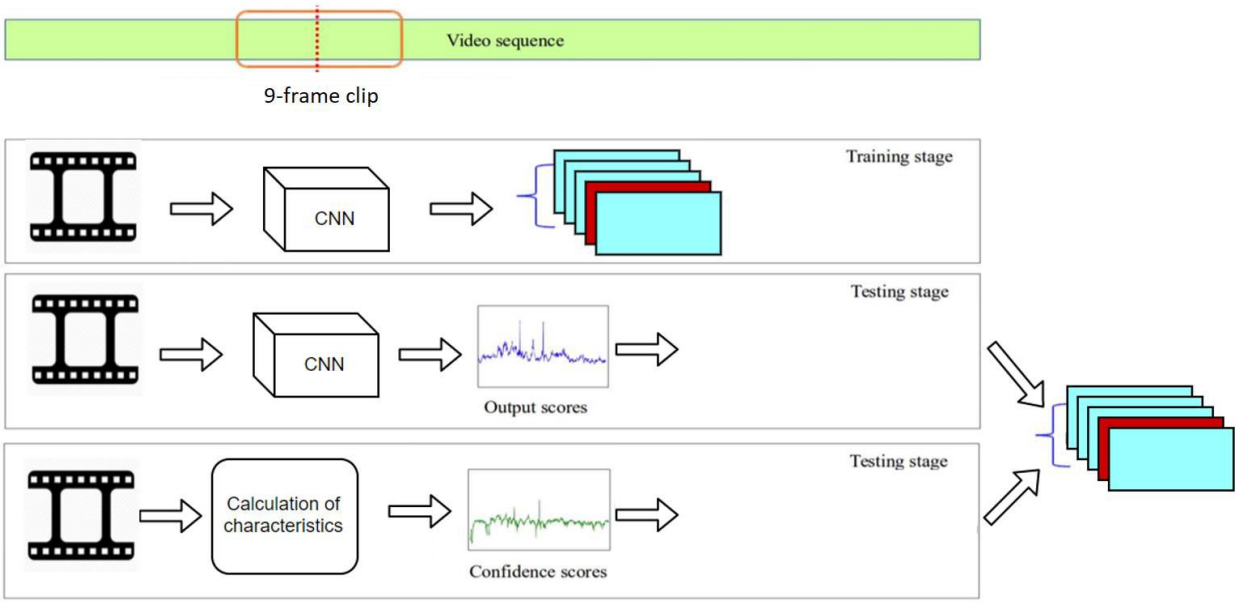

Figure 2. The pipeline of the proposed method.

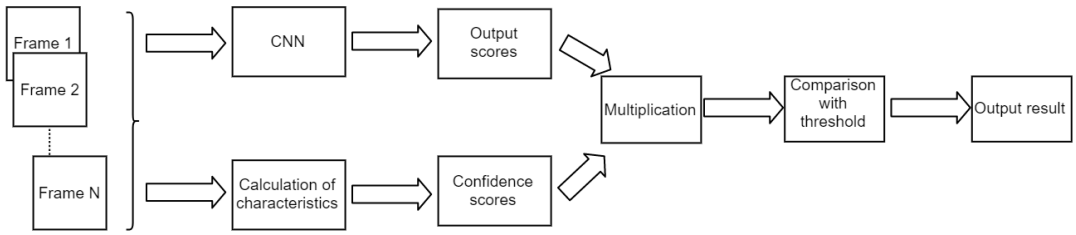

Figure 3. The algorithm workflow.

At the first step we calculate *ConfidenceScores* using the set statistical characteristics for each pair of adjacent frames f_k=*frgb(:,:,k-1)* and f_{k-1}=*frgb(:,:,k)*:

1) The inter-frame difference:

$$DIFf_k = sum(sum(abs(f_k - f_{k-1})),$$

2) The inter-frame difference of mathematical expectations:

$$DIFfMat_k = abs(fMat_k - fMat_{k-1}),$$

$$fMat = \frac{1}{SizeRow * SizeColumn} \sum_{i=1}^{SizeRow} \sum_{j=1}^{SizeColumn} f_{i,j},$$

3) The inter-frame difference of variance:

$$DIFfDisp_k = \text{abs}(fDisp_k - fDisp_{k-1}),$$

$$fDisp = \frac{1}{SizeRow * SizeColumn} \sum_{i=1}^{SizeRow} \sum_{j=1}^{SizeColumn} (f_{i,j}(k) - fMat_k)^2,$$

4) The inter-frame difference of brightness histograms:

$$DIFfHist_k = sum(abs(fHist_k - fHist_{k-1})),$$

5) The correlation coefficient between the frame f_{k-1} and the compensated frame $fComp_k$:

$$fCor_k = corr(f_{k-1}, fComp_k),$$

$$fCor_k = \frac{\sum_{i=1}^{SizeRow}\sum_{s=1}^{SizeColumn}(f_{i,j}(k)-MfMat_k)(fComp_{i,j}(k)-fCompMat_k)}{\sqrt{\sum_{i=1}^{SizeRow}\sum_{s=1}^{SizeColumn}(f_{i,j}(k)-fMat_k)^2\sum_{i=1}^{SizeRow}\sum_{s=1}^{SizeColumn}(fComp_{i,j}(k)-fCompMat_k)^2}},$$

6) The difference between the frame f_{k-1} and the compensated frame $fComp_k$:

$$DIFfComp_k = sum(sum(abs(f_{k-1} - fComp_k))),$$

7) The inter-frame difference of mathematical expectation of the optical flow:

$$DIFfOptMat_k = abs(fOptMat_k - fOptMat_{k-1}),$$

8) The inter-frame difference of variance of the optical flow:

$$DIFfOptDisp_k = abs(fOptDisp_k - fOptDisp_{k-1}),$$

9) The inter-frame difference of standard deviation of the optical flow:

$$DIFfOptMSE_k = abs(\sqrt{fOptDisp_k} - \sqrt{fOptDisp_{k-1}}),$$

10) The correlation coefficient between the amplitude of the optical flow and the compensated amplitude of the optical flow:

$$fOptCor_k = corr(fOpt_{k-1}, fOptComp_k),$$

11) The difference between the amplitude of the optical flow and the compensated amplitude of the optical flow:

$$DIFfOptComp_k = sum(sum(abs(fOpt_{k-1} - fOptComp_k))),$$

$$MedFilt = medfilt1(h, t),$$

$$h = (DIFfMat_k, DIFfDisp_k, DIFfHist_k, fCor_k, DIFfComp_k, DIFfOptMat_k, DIFfOptDisp_k, DIFfOptMSE_k, fOptCor_k, DIFfOptComp_k),\ t = 5.$$

$$DIFMedDIFfMat_k = abs(DIFfMat_k - MedDIFfMat_k),$$

$$DIFMedDIFfDisp_k = abs(DIFfDisp_k - MedDIFfDisp_k),$$

$$DIFMedDIFfHist_k = abs(DIFfHist_k - MedDIFfHist_k),$$

$$DIFMedDIFfCor_k = abs(DIFfCor_k - MedDIFfCor_k),$$

$$DIFMedDIFfComp_k = abs(DIFfComp_k - MedDIFfComp_k),$$

$$DIFMedDIFfOptMat_k = abs(DIFfOptMat_k - MedDIFfOptMat_k),$$

$$DIFMedDIFfOptDisp_k = abs(DIFfOptDisp_k - MedDIFfOptDisp_k),$$

$$DIFMedDIFfOptMSE_k = abs(DIFfOptMSE_k - MedDIFfOptMSE_k),$$

$$DIFMedDIFfOptCor_k = abs(DIFfOptCor_k - MedDIFfOptCor_k),$$

$$DIFMedDIFfOptComp_k = abs(DIFfOptComp_k - MedDIFfOptComp_k),$$

$$NorDIFf_k = \frac{DIFf_k}{max(DIFf_k)},$$

$$NorDIFMedDIFfMat_k = \frac{DIFMedDIFfMat_k}{max(DIFMedDIFfMat_k)},$$

$$NorDIFMedDIFfDisp_k = \frac{DIFMedDIFfDisp_k}{max(DIFMedDIFfDisp_k)},$$

$$NorDIFMedDIFfHist_k = \frac{DIFMedDIFfHist_k}{max(DIFMedDIFfHist_k)},$$

$$NorDIFMedDIFfCor_k = \frac{DIFMedDIFfCor_k}{max(DIFMedDIFfCor_k)},$$

$$NorDIFMedDIFfComp_k = \frac{DIFMedDIFfComp_k}{max(DIFMedDIFfComp_k)},$$

$$NorDIFMedDIFfOptMat_k = \frac{DIFMedDIFfOptMat_k}{max(DIFMedDIFfOptMat_k)},$$

$$NorDIFMedDIFfOptDisp_k = \frac{DIFMedDIFfOptDisp_k}{max(DIFMedDIFfOptDisp_k)},$$

$$NorDIFMedDIFfOptMSE_k = \frac{DIFMedDIFfOptMSE_k}{max(DIFMedDIFfOptMSE_k)},$$

$$NorDIFMedDIFfOptCor_k = \frac{DIFMedDIFfOptCor_k}{max(DIFMedDIFfOptCor_k)},$$

$$NorDIFMedDIFfOptComp_k = \frac{DIFMedDIFfOptComp_k}{max(DIFMedDIFfOptComp_k)}.$$

Next, the $Confidence\ Scores$ is calculated as:

$$ConfidenceScores_k = (NorDIFMedDIFfMat_k + NorDIFMedDIFfDisp_k + NorDIFMedDIFfHist_k + NorDIFMedDIFfCor_k + NorDIFMedDIFfComp_k + NorDIFMedDIFfOptMat_k + NorDIFMedDIFfOptDisp_k + NorDIFMedDIFfOptMSE_k + NorDIFMedDIFfOptCor_k + NorDIFMedDIFfOptComp_k)/ \max(NorDIFMedDIFf_k).$$

The second step of the proposed algorithm is the detection of deleted frames $OutputScores$ on videos using the Convolutional Neural Network. The CNN defines the class, to which each frame. The architecture of the neural network is shown in Fig. 4. The model has following parameters were used to train for all experiments: size of the mini batch equal 40, hidden convolutional layers produce 30, 50 and 70 feature maps with a kernel size of 3×3 pixels, respectively, first fully connected layer has 280 neurons, the learning rate is 0,0001. It is important to note that the threshold value for all experiments is 0,7. The minimum classification error was achieved on average after 200 epochs.

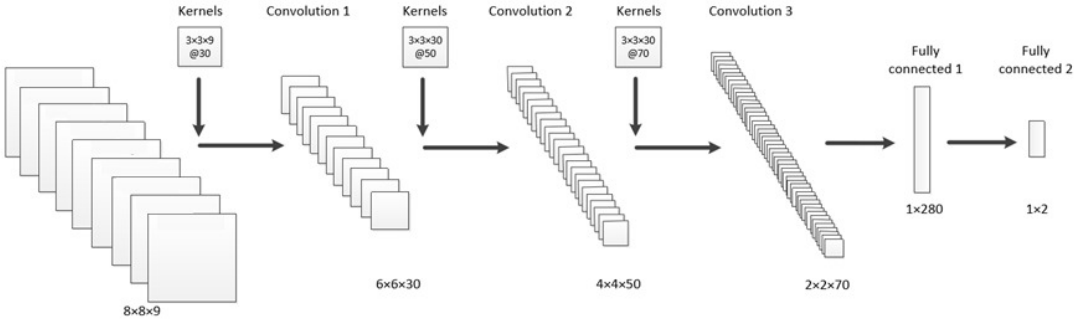

Figure 4. The architecture of the proposed convolutional network.

At the final stage, the vectors $ConfidenceScores_k$ and $OutputScores_k$ multiply element by element to form the resulting vector $Result_k$:

$$Result_k = ConfidenceScores_k * OutputScores_k,$$

The time of a deleted frame is detected if any of the values in the $Result_k$ vector exceeds the threshold value $Threshold_k$, which is calculated as:

$$Threshold_k = \frac{Max+Min}{2},$$

$$Max = max(Result_k),$$

$$Min = min(Result_k),$$

3. EXPERIMENTAL RESULTS

To obtain the training data we received 3000 videos. The length all videos are of 1-3 minutes. Some of the frames from this dataset presents in Figure 5. All videos deviated on two groups: light conditions or complex conditions. The light condition is the contrast videos with light brightness and slow object motion. The videos at the group with complex conditions include lack of contrast, irregular lighting and brightness images what may not preserve the local image features/details. Some of the videos are stationary scene and not contains moving objects.

We randomly selected 300 videos for training and adopted the rest 50 videos for validation. We developed a tool that randomly drops fixed length frame sequences from videos. In our experiments, we manipulate each video many different times to create more data. We vary the fixed frame drop length to see how it affects detection we used 0.5s, 1s, 2s, as different frame drops durations.

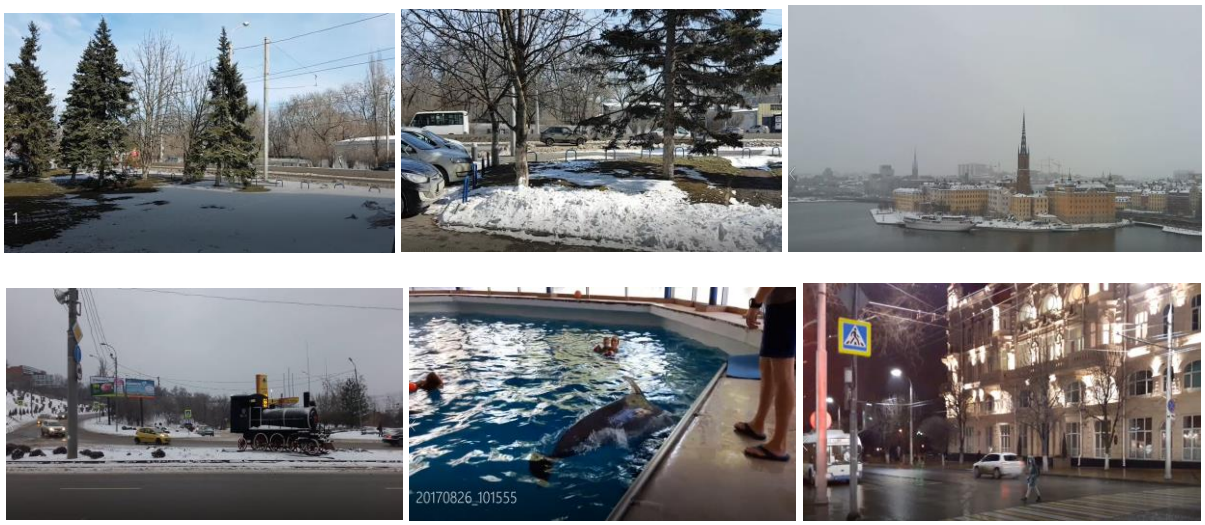

Figure 5. Frames from test dataset.

To evaluate the effectiveness of the proposed method we use the following metrics:

- The probability of correct detection:

$$Prob.CorrectPrediction = \frac{TP}{NumDefPix}$$

- The probability of a false alarm:

$$Prob.FalseAlarm = \frac{FP}{NumAllPix - NumDefPix}$$

- The probability of false missing:

$$Prob.FalseMissing = \frac{FN}{NumAllPix - NumUndmgPix}$$

where *TP* is the true positive, *FP* is false positive, *FN* is a false negative, *NumDefPix* - the number of pixels belonging to a crack, *NumAllPix* - total number of pixels, *NumUndmgPix* – the number of pixels not belonging to the crack.

The results of calculations the probabilities shown in Table 1. The analysis of the obtained results indicates that the efficiency of the developed method is quite high and that the use of neural networks leads to significantly reduced probabilities of false alarms.

Table 1. The probabilities for detection of deleted frames on videos.

	The probability of false alarm	The probability of false missing	The probability of correct detection
Light conditions	3,8%	3,8%	96,1%
Complex conditions	8,5%	16%	88,9%

CONCLUSIONS

We propose the approach for the detection of deleted frames in videos. The proposed algorithm is a two-stage procedure: the statistical analysis and the convolutional neural network. We calculate the set of different statistical rules for all frames as confidence scores. Also, the convolutional neural network used to obtain the output scores. Based on the two score curves, we calculation multiplication between them and use threshold for detection deleted frames for per frame clip. The proposed method can identify whether frame dropping exists and even determine the exact location of the frame is dropping without any information of the reference/original video. Experimental results demonstrate the effectiveness of the proposed approach to a test video database. In the future, there is planned to apply the presented approach to the video sequence in real time, to make comparisons with state-of-the-art methods.

ACKNOWLEDGMENT

This work was supported by PROVER project (https://prover.io/).

ABN thanks the Ministry of Science and Education for support under grants 1.6198.2017/6.7 and 1.7706.2017/8.9 and Center of Collective Use of MSTU Stankin for providing resources.

REFERENCES

[1] Redi, J. A., Taktak, W., and Dugelay, J. L., "Digital image forensics: a booklet for beginners," Multimed Tools Appl, Vol. 51(1), 133–162, (2011).

[2] Wang, W., "Digital video forensics," Ph.D. dissertation. Department of Computer Science, Dartmouth College, Hanover, New Hampshire, (2009).

[3] Subramanyam, A.V. and Emmanuel, S., "Video forgery detection using HOG features and compression properties," in Proc. IEEE 14th International Workshop on Multimedia Signal Processing (MMSP 2012), 89-94, (2012).

[4] Upadhyay, S. and Singh, S. K., "Learning Based Video Authentication using Statistical Local Information," in Proc. International Conference on Image Information Processing (ICIIP 2011), 1-6, (2011).

[5] Yu, J. and Srinath, M.D., "An efficient method for scene cut detection," Pattern Recognition Letters, 22, 1379-139, (2001).

[6] Yusoff, Y., Christmas, W., and Kittler, J., "Video Shot Cut Detection Using Adaptive Thresholding," in Proc. British Mission Vision Conference (BMVC), 11-14, (2000).

[7] Muhammad, G., Hussain, M., and Bebis, G., "Passive copy move image forgery detection using undecimated dyadic wavelet transform," Digital Investigation, 49–57, (2012).

[8] Chetty, G., Biswas, M., and Singh, R., "Digital Video Tamper Detection Based on Multimodal fusion of Residue Features," in Proc. 4th International Conference on Network and System Security (NSS), 606-613, (2010).

[9] Hong, J.Z., Kankanhalli, A., Smoliar, S.W., "Automatic partitioning of full-motion video," Multimedia Systems, (1993).

[10] Yu, J., Srinath, M.D., "An efficient method for scene cut detection," Pattern Recognition Letters, 1379-1391, (2001).

[11] Kapur, J.N., Sahoo, P.K., Wong, A.K.C., "A new method for gray-level picture thresholding using the entropy of the histogram," Computer vision, graphics, and image processing, 273-285, (1985).

[12] Zhenzhen, Z., Jianjun, H., Qinglong, M., Zhaohong, L., "Efficient video frame insertion and deletion detection based on inconsistency of correlations between local binary pattern coded frames," Security Comm. Networks, 311-320, (2015).
[13] Pietikäinen, M., Ojala, T., Maenpaa, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns" IEEE Transactions on pattern analysis and machine intelligence, vol. 24(7), (2002).
[14] Voronin, V., Marchuk, V., Sizyakin, R., Gapon, N., Pismenskova, M., Tokareva, S., "Automatic image cracks detection and removal on mobile devices," Proc. SPIE 9869, Mobile Multimedia/Image Processing, Security, and Applications, 98690R, (2016).
[15] Pietikäinen, M., Ojala, T., "Texture analysis in industrial applications," IT Advances in Image Processing, Multimedia and Machine Vision, 337-359, (1996).