Classification of Video Recaptured from Display Device

Minoru Kuribayashi[1], Kodai Kamakari, Kento Kawata, Nobuo Funabiki
Okayama University, Okayama, Japan
E-mail: [1] kminoru@okayama-u.ac.jp

***Abstract*—The prevention from unauthorized recapturing of screen is an important issue in multimedia security. In this study, we attempt to detect illegally created videos captured from display devices by analyzing unnatural signals contained in the videos. The proposed approach applies a convolutional deep neural network (CNN) for the classification. In order to reduce the computational costs, some frames are sampled from a target video, and are checked whether they are captured. In the training process, each frame sampled from captured/natural videos is partitioned into small patches, and a CNN model is trained by using the patches. The final decision is determined from the classification results at each frame. We conducted experiments to evaluate the classification accuracy and its dependency on camera devices. It is confirmed that we can classify captured and natural videos with high probability under our experimental conditions. When a same camera device is used for recording both original and recaptured videos, the classification accuracy is decreased from the case of different devices.**

I. Introduction

Along with the evolution of various electric appliances, the camera devices are remarkably miniaturized without sacrificing the quality of recorded content. It is easy for us to capture digital images and videos with devices such as smartphones and digital cameras including action cameras optimized for shooting outdoor sports. Although those camera devices are convenient, there are a lot of issues to be solved due to the illegal usage of the devices. One of them is a voyeur of a movie. A movie industry has suffered a great deal of damage due to the unauthorized outflow of a large number of duplicates. In addition, a wide-band Internet access enables malicious users to upload/download the unauthorized copies of video to social network platforms and video streaming services.

One of the promising techniques is the digital watermarking [1], [2], [3] which embeds sub-information into images and video so that the ownership of the content can be claimed. The other techniques are multimedia forensics which analyze some traces induced by operations in hardware and software [4], [5]. When different camera devices capture a same scene, the image/video contains different patterns of distortions, which is hardware-oriented noise. The difference of encoding algorithms also may creates different distortions in the recorded content. In the past studies on forensics, several algorithms based on statistical analysis and pattern recognition have been investigated [6], [7]. The invention of new machine learning technique called deep learning provides great improvements in various applications of pattern recognition system. In conventional works [8], [9], [10], [11], convolutional neural network(CNN)-based techniques for classifying recaptured and natural images have been investigated. At first, an input image is usually partitioned into small and fixed-size pixel patches. Then, all patches are processed, or a patch selection strategy can be applied to choose the patches that are more useful for the following task. For each patch, a preprocessing operation is performed to remove unnecessary signals for classification and a CNN model outputs its classification result. Finally, the classification result for the input image is determined by summarizing the results among the patches. It is possible to reduce the computational costs by selecting some useful patches under predefined conditions.

The similar techniques for recapturing image forensics can be applied to the images created by computer graphics [12], [13], [14]. Rahmouni et al. [15] presented a novel statistical features extraction layer, and embedded it between the last convolutional layer and the first fully connected layer. As the target of the above-mentioned methods is still images, the extension to whole video is not discussed in details.

In this study, we try to detect illegal videos created by capturing a display. In the proposed method, some frames of a target video are sampled, and each frame is further divided into patches of 100×100 pixels. We train a CNN-based classifier with respect to the patches, and determine the decision for each frame based on the classification results of the patches. According to the decision of all sampled frames, the final decision is determined. We employ the method in [15] for the basic tool as the classifier, and investigate the sensitivities of difference in the image size of frames and camera devices.

First, we examine each patch and calculate the probability that it is created by recapturing a screen. Next, we summarize the probabilities of all patches in one sampled frame to determine whether it is recaptured or not. The final decision is done for a given video according to the results of all sampled frames. To evaluate the proposed framework for the detection of recaptured videos, we conduct experiments for some videos selected from a public library and their recaptured versions by using some video devices.

II. Image Forensics

There are many studies about image forensics. Some techniques are based on statistical analysis in various characteristics observed from maliciously modified images. Among many branches in the techniques, we focus on the detection of

recaptured image in this paper. The motivation of recaptured image forensics is to find the images generated by capturing a printed picture or a screen display with an acquisition device. The main task is to answer whether an image has been recaptured or not.

A. Conventional Works

The first work to detect recaptured images based on deep learning is the method in [8]. In the method, the Laplacian filter is embedded into the first layer of a CNN to improve the noise signal ratio introduced by recapture operations. In [10], the convolutional operation is introduced as the preprocessing. Features extracted from trained CNN model were then fed into a recurrent neural network to classify the images. For evaluating recaptured image forensic techniques, a large dataset is prepared for experiments in [11], and some different kinds of Gaussian filtering residuals are also introduced in the first layer to improve the accuracy. As an image's subject is not useful for the classification, those filters attempts to discard such information while the traces induced by the recapturing operation is enhanced.

An interesting branches of image forensics is the classification of computer-generated images and natural images. In [12], VGG-based architectures are evaluated for computer-generated image detection. It is showed that their performance could be improved by dropping max-pooling layers. Some common CNN-based architectures such as VGG-19 and ResNet50 are evaluated in [16], and the effects of fine-tuning and transfer learning techniques are measured. More complicated architectures combining CNN and RNN are investigated in [10], [13]. Instead of using fixed filters in the preprocessing step, several convolutional operations are employed in [14]. Rahmouni et al. [15] presented a CNN-based system with a statistical features extraction layer which extract four features: mean, variance, maximum, and minimum.

B. CG vs Photo

"CGvsPhoto"[1] is developed by Rahmouni et al. in [15], and the source code is available on GitHub.it. It implements a program that classifies computer graphics(CG) images and photographs(Photo) using a CNN with TensorFlow as the back end. The procedure to train and test the accuracy of the CGvsPhoto tool is summarized as follows:

1) Database creation
   It collects labeled images and randomly select 500 images for CG and Photo, respectively. Furthermore, each 500 images are randomly divided into 350 images for training data, 50 images for validation data, and 100 images for test data.
2) Creating a patch database
   The selected 500 images are divided into patches of 100×100 pixels each. For example, if an input image is 400×300 pixels, it is divided into 4×3 patches. Among such divided patches, 20,000 patches are randomly selected as a training data, 2,000 and 4,000 are selected as the validation data and the test data, respectively.
3) Model training
   The training of a CNN model is performed using 10,000 patches randomly selected from the training data. In addition, the validation of the model is done with 40 patches in the validation data for every 100 epochs of the training, and the accuracy of the model at that time is evaluated. The 40 patches are randomly selected from 2000 patches in the validation data. After the training, the model is evaluated with 80 patches of test data.
4) Model testing
   The classification result for an input image is determined by voting among the patches. At this time, the judgment results of each patch are summed up to comprehensively judge the original image. There are the following two methods for discriminating one original image from the judgment result for each patch.
   In *Majority Voting*, the binary classification results of patches are counted, and the one that exceeds the majority is used as the judgment result for each image. On the other hand, *Weighted Voting* calculates the probability for each divided patches, and sums up them to obtain the overall score taken as the judgement.

[1] https://github.com/NicoRahm/CGvsPhoto

III. Proposed Method for Recaptured Video Forensics

The objective of this study is to detect the videos created by recapturing screen. In general, there is a problem that a huge amount of calculation is required to analyze the video directly, and the program may not operate normally depending on the storage format of the video to be detected. As a solution to this problem, we first extract some frames from a given video, and try to classify each frame by using a conventional method.

A. Classifier

As discussed in Section II-A, there are some reports that filtering operations are useful to enhance the classification accuracy in recaptured image forensics. Regretfully, conventional methods [8], [10], [11] developed their classifiers using own CNN-based architectures. On the other hand, in the forensics for computer-generated images [12], [13], [14], common architectures such as VGG and ResNet are employed as the basis of their classifiers, which is easier to expand their techniques to the other applications. Even though such classifiers are developed for computer graphics, the techniques retain the similar characteristics in the recaptured image forensics. Once the training dataset of recaptured images and natural images are given, high classification accuracy can be achieved. Especially the CGvsPhoto tool, it is easy to transplant the technique to the recaptured image forensics as its source code is available at GitHub.it. In addition, the statistical features extraction layer can work effectively to the recaptured images as well as computer graphics. Hence, we employ the CGvsPhoto tool

as the basic classifier for frames sampled from a target video in our method.

B. Identification procedure

First, the probability of the 100×100 patch being the patch of the recaptured image (or the patch of the original image) is calculated. Next, one image is classified whether it is a recaptured image or an original image based on the above calculation results. At this time, in the Majority Voting, an erroneous judgment may occur when there is a local bias in the probability. Therefore, the Weighted Voting is adopted in this research. Then, the recapturing video and the original video are finally identified by following the same procedure for multiple images.

C. Dataset

If a similar set of scenes are involved in the videos for training a classifier, the training may not be performed properly. In addition, if the videos used in the training were similar to the videos used in the test, the accuracy may not be measured accurately. Therefore, it is desirable to collect various scenes of videos for a dataset. Among the videos in such a dataset, several frames are sampled for training the classifier.

One of the candidates for the dataset is to download a public library. In addition to the videos in the library, we create some videos using some camera devices. For the original videos, we also use the camera devices to create their recaptured versions.

Assume that there are n videos of different scenes. Using m different camera devices, we create m versions of recaptured for each video. In total, mn recaptured videos are available for experiments in this setting.

If the original videos are selected from a library, the original camera device is unknown. In general, the high quality video such as movies and TV content are created by using an expensive camera device, while a recapturing camera device is much cheaper. So, we assume that the different camera devices are used for original and recaptured videos. Under such an assumption, we train a classifier and evaluate the classification accuracy in the following experiments.

IV. Experiments

We conduct experiments under the environments summarized in Table I. The number of camera devices is $m = 2$. One is the EverioR GZ-RX680 which is a representative of a normal video camera and the other is GoPro HERO7 SILVER which is a representative of a small high-performance camera.

The NHK Creative Library[2] is used as the original videos. Among several videos in the library, $n = 10$ archives of 30 to 90 seconds are selected; one for birds, fish, dinosaurs, aerial photographs of Sapporo, food, and humans, and two for natural landscapes and townscapes. Their snapshot of a certain frame is shown in Fig. 1. A movie of about 700 seconds was created by concatenating all of these movies.

In order to investigate the dependency on the camera devices in the original video and the recaptured one, we made short videos of about 300 seconds using EverioR and GoPro by taking the scene of campus in Okayama University while walking through. The videos were recorded around 2 pm in Japan time on sunny day in December. Image quality is determined by the combination of bit rate and type of codec. The bit rate is a value of how much data in one second of video. The type of codec represents a video or sound compression algorithm. The codecs of video is H.264 both for EverioR and GoPro while their bit rates are about 17 Mbps and 60 Mbps for EverioR and GoPro, respectively. The snapshot of such videos are shown in Fig. 2. The frame rate of the videos is 30fps and each frame size is full HD 1920×1080 pixels unless otherwise specified.

In this experiment, we created three kinds of original videos. One is the video which is concatenated 10 archives in the NHK Creative Library. The other two videos are captured by using EverioR and GoPro camera. For these videos, we create recaptured videos using these two cameras, and train a classifier with the frames of original and recaptured videos as supervised data.

TABLE I
Evaluation environment

CPU	AMD Ryzen 7 2700X
RAM	32GB (DDR4-2666)
GPU	Nvidia GeForce RTX2080 (8GB)
Camera Device	EverioR GZ-RX680 GoPro HERO7 SILVER
monitor	PHILIPS 274E SoftBlue
Video editing software	Aviutl
Major software/library	TensorFlow-gpu 1.14.0 TensorBoard 1.14.0 CUDA 10.2 Python 3.6.8

Fig. 1. Snapshot of videos in the NHK Creative Library.

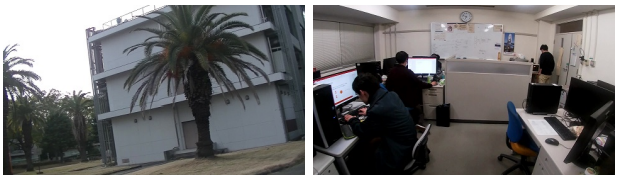

Fig. 2. Snapshot of the original videos taken by Everio R or GoPro?.

[2]http://www1.nhk.or.jp/archives/creative/material/

A. Training Phase

Due to the convenience, each frame in a given video is regarded as a suspicious image to be evaluated by a classifier "CGvsPhoto". In this experiment, we train the CNN model in the classifier with default parameters using the patches randomly selected from the respective folders for training, validation and test.

At the training, frames are sampled from a given video for every 1 second in case of the NHK Creative Library, and each frames are partitioned into non-overlapped patches with 100×100 pixels. As the size of frames is 1920×1080 pixels, 190 patches are obtained for each frame. Hence, for a 100-seconds video, 190000 patches are obtained. In case of other two videos, frames are sampled for every 0.5 second.

We collected extract frames original videos, and randomly selected 500 frames, and divided them into three groups: training, validation, and test. Similarly, 500 frames are selected from recaptured videos. For each sampled frame, we randomly put it into one folder among three groups: training, validation, and test. For training, there are 350 frames with labels of original and recaptured, respectively, and they are divided into 66500 (=350×190) patches. Similarly, 50 and 100 frames are selected for validation and test from respective folders so as not to use the patches from a same frame at training, validation, and test phase.

B. Evaluation Index

Since we study a binary classification task that classifies the frames of original videos and the recaptured videos, there are a total of 4 patterns of true or false as a result of predicting that the correct answer data is positive or negative. Each result is expressed as true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Each case is explained in detail.

- TP Frames which are created by recapturing screen and actually estimated recaptured one.
- TN Frames which are originals and actually estimated original one.
- FP Frames which are originals and actually estimated recaptured one.
- FN Frames which are created by recapturing screen and actually estimated original one.

When training the CNN model, the accuracy of the model is evaluated with 40 patches in validation data every 10 training epochs. In this study, the correct answer rate at this time is simply called "Accuracy". The definition formula of the Accuracy is shown below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

After the model training is completed, the model is evaluated using 80 patches in test data. The accuracy for the patch level is called "patch accuracy". For each frame, we evaluate the accuracy by using the Weighted Voting, which is called "frame accuracy".

TABLE II
CLASSIFICATION ACCURACY FOR THE VIDEOS IN NHK CREATIVE LIBRARY.

	Accuracy(%)	
	patch	frame
EverioR	71.3	94.0
GoPro	82.6	99.2

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY FOR DIFFERENT FRAME SIZE USING GOPRO.

	Accuracy(%)	
frame size	patch	frame
1920×1080	82.6	99.2
1280×720	78.1	94.3
640×480	81.6	90.4

C. Classification Accuracy

In this subsection, we use the videos in the NHK Creative Library as the original. The classification accuracy of original videos and recaptured ones are evaluated for EverioR and GoPro. Table II shows the patch accuracy and the frame accuracy. Despite recapturing the same original video, differences in classification accuracy appeared depending on the recapturing device. Nevertheless, as the frame accuracy is sufficiently high, it can be said that almost all frames could be classified as original or recaptured. Since various scenes are included in the library, it is considered possible to classify it regardless of the content of the video.

Next, we investigate whether there is a difference in classification accuracy depending on the frame size. The classification accuracy is compared for the frames of 1920×1080, 1280×720, and 640×480 pixels for the frames of original videos and the videos recaptured by GoPro.Table III shows the patch accuracy and the frame accuracy for these different frame sizes. From the table, it is observed that the patch accuracy is almost constant regardless of the frame size, while the frame accuracy decreases as the frame size decreases. This is because the patch accuracy is judged in the patch unit of 100×100 pixels regardless of the frame size. On the other hand, regarding the frame accuracy, a frame of 1920×1080 pixels is composed of 190 ($= 19 \times 10$) patches, while a frame of 640×480 pixels is composed of only 24 ($= 6 \times 4$) patches. The drop of the performance comes from the insufficient number of patches for classification.

D. Dependency of Camera Device

In order to investigate the dependency of camera device on the classification accuracy, two kinds of videos are created by using EverioR and GoPro. Assuming these videos are original, an experiment is conducted using two cameras. There are four cases in this setup as enumerated in Table IV.

For each case, we train a classifier by using the frames of the corresponding original and recaptured videos. The results are shown in Table V. From the table, it is found that when the original video and the recapturing video are created by a same camera device, the classification accuracy dropped

TABLE IV
COMBINATION OF CAMERA DEVICES FOR CREATING ORIGINAL AND RECAPTURED VIDEOS.

case	Original	Recaptured
i	EverioR	GoPro
$\ddot{u}$	EverioR	EverioR
$\ddot{\ddot{m}}$	GoPro	GoPro
$\dot{w}$	GoPro	EverioR

TABLE V
DEPENDENCY OF CAMERA DEVICES ON THE CLASSIFICATION ACCURACY.

	Accuracy(%)	
case	patch	frame
i	99.8	100.0
$\ddot{u}$	69.0	83.1
$\ddot{\ddot{m}}$	69.8	88.3
$\dot{w}$	98.7	99.5

significantly. On the other hand, if the recaptured video is created by using a different camera device, we can obtain remarkably high accuracy as shown in the cases i and $\dot{w}$. One of the reason is the difference of hardware in the device such as optical characteristic of lens, color filter, sensor device, and so on. In general, the videos to be protected are recorded by using an expensive devices, while an illegally video is recaptured by using a cheaper device. Such a difference in the device will increase the opportunity for us to detect the recaptured video with high probability.

V. CONCLUSIONS

In this study, we investigated the classification of recaptured videos and the dependency of the classification accuracy on the difference of devices. At the classification of the frames of original and the recaptured videos, we can obtain the frame accuracy exceeded 99% when the video is recaptured by using the different device from the one used for original video. Under our experimental condition, it was confirmed that the classification accuracy deteriorates when the frame size is small. One of our future works is to conduct more experiments for various devices and environments.

ACKNOWLEDGMENT

This research has been partially supported by the JSPS KAKENHI Grant Number 19K22846.

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, 2001.
[2] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*, Signal Processing and Communications, Marcel Dekker, 2004.
[3] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital WaterMarking and Steganography*, Morgan Kaufmann, 2008.
[4] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, no. 496701, pp. 22, 2013.
[5] P. Yang, D. Baracchi, R. Ni, Y. Zhao, F. Argenti, and A. Piva, "A survey of deep learning-based source image forensics," *MDPI J. Imaging*, vol. 6, no. 9, pp. 24, 2020.
[6] H. Farid, "Image forgery detection," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, 2009.
[7] B. Mahdian and S. Saic, "A bibliography on blind methods for identifying image forgery," *Signal Processing: Image Communication*, vol. 25, no. 6, pp. 389–399, 2010.
[8] P. Yang, R. Ni, and Y Zhao, "Recapture image forensics based on laplacian convolutional neural networks," in *Proc. IWDW'16*, 2017, vol. 10082, pp. 119–128.
[9] H. Y. Choi, H. U. Jang, J. Son, D. Kim, and H. K. Lee, "Content recapture detection based on convolutional neural networks," in *Proc. ICISA'17*, 2017, pp. 339–346.
[10] H. Li, S. Wang, and A. C. Kot, "Image recapture detection with convolutional and recurrent neural networks," *Electronic Imaging, Media Watermarking, Security, and Forensics*, vol. 5, pp. 87–91, 2017.
[11] S. Agarwal, W. Fan, and H. Farid, "A diverse large-scale dataset for evaluating rebroadcast attacks," in *Proc. ICASSP'18*, 2018, pp. 1997–2001.
[12] I. J. Yu, D. G. Kim, J. S. Park, J. U. Hou, S. Choi, and H. K. Lee, "Identifying photorealistic computer graphics using convolutional neural networks," in *Proc. ICIP'17*, 2017, pp. 4093–4097.
[13] P. He, X. Jiang, T. Sun, and H. Li, "Computer graphics identification combining convolutional and recurrent neural networks," *IEEE Signal Process. Letters*, vol. 25, pp. 1369–1373, 2018.
[14] W. Quan, K. Wan, D. M. Yan, and X. Zhang, "Distinguishing between natural and computer-generated images using convolutional neural networks," *IEEE Trans. Information Forensics and Security*, vol. 18, pp. 2772–2787, 2018.
[15] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *Proc. WIFS'17*, 2017, pp. 1–6.
[16] M. He, *Distinguish computer generated and digital images: A CNN solution*, Concurrency Computing, Pract. Exp., 2018.