Digital Watermarking Method for Printed Matters Using Deep Learning for Detecting Watermarked Areas

Hiroyuki IMAGAWA[†], *Nonmember*, Motoi IWATA^{†a)}, *Senior Member*, and Koichi KISE^{†b)}, *Fellow*

SUMMARY There are some technologies like QR codes to obtain digital information from printed matters. Digital watermarking is one of such techniques. Compared with other techniques, digital watermarking is suitable for adding information to images without spoiling their design. For such purposes, digital watermarking methods for printed matters using detection markers or image registration techniques for detecting watermarked areas are proposed. However, the detection markers themselves can damage the appearance such that the advantages of digital watermarking, which do not lose design, are not fully utilized. On the other hand, methods using image registration techniques are not able to work for non-registered images. In this paper, we propose a novel digital watermarking method using deep learning for the detection of watermarked areas instead of using detection markers or image registration. The proposed method introduces a semantic segmentation based on deep learning model for detecting watermarked areas from printed matters. We prepare two datasets for training the deep learning model. One is constituted of geometrically transformed non-watermarked and watermarked images. The number of images in this dataset is relatively large because the images can be generated based on image processing. This dataset is used for pre-training. The other is obtained from actually taken photographs including non-watermarked or watermarked printed matters. The number of this dataset is relatively small because taking the photographs requires a lot of effort and time. However, the existence of pre-training allows a fewer training images. This dataset is used for fine-tuning to improve robustness for print-cam attacks. In the experiments, we investigated the performance of our method by implementing it on smartphones. The experimental results show that our method can carry 96 bits of information with watermarked printed matters.

key words: digital watermarking, deep learning, semantic segmentation, print-cam attack, mobile device

1. Introduction

Today mobile devices with a camera such as a cellphone and a smartphone are very popular. The techniques to get digital information from taken images by mobile devices are developing. The most famous one is a QR code [1]. We can get a lot of information from a QR code quickly. However, QR codes degrade design due to their white and black patterns with detection markers. In the case add information to images without degrading design, using digital watermarking is one of the best approaches. Digital watermarking is a technique for embedding any information into digital contents can be imperceptible. For extracting information from taken watermarked images, the detection of watermarked ar-

eas is a challenging task. Some watermarking methods for getting digital information from printed images by mobile devices have been proposed.

Nakamura, et al. [2] and Pramila, et al. [3], [4] proposed the digital watermarking method that can extract a watermark from a printed image with the camera of a cellular phone. In these methods, a watermarked image was surrounded by a black frame border. Yamanaka, et al. [5] proposed the digital watermarking method using a finder pattern and an alignment pattern used in QR codes for detecting the position of watermarked images. These methods restrict or spoil the design of images by detection markers in exchange for the easy detection of watermarked areas.

To solve this problem, Ogawa et al. [6] proposed the digital watermarking method that can get information from a printed watermarked image without adding a position detection marker based on an image retrieval. Zhong, et. al. [7] also tried to detect watermarked areas based on an image registration technique. These methods have a disadvantage that watermarks cannot be extracted from non-registered images because image retrieval can match only registered images. It causes inconvenience.

In this paper, we propose a novel digital watermarking methods using deep learning for detecting watermarked areas without registering original images or using detection markers. The task of deep learning is judging whether a watermarked area or not by pixel-wise as semantic segmentation.

In Sect. 2, the proposed method is explained. In Sect. 3, the datasets for training a deep learning model are described. Experiments to analyze the performance of our method on a mobile device are given in Sect. 4. Finally, the conclusions are drawn in Sect. 5.

2. Proposed Method

The proposed digital watermarking method for printed matters is composed of two parts, that is, watermark embedding and watermark extraction.

2.1 Watermark Embedding

Figure 1 shows the overall of watermark embedding. Let $I = (I_{x,y}^R, I_{x,y}^G, I_{x,y}^B)$ ($0 \leq x < W, 0 \leq y < H$) be the original color image. The 8K bits watermark information is encoded by Reed Solomon coding to coded bits $c = \{c_i | c_i \in \{0, 1\}, 0 \leq i < N^2\}$. Next, basic 2-dimensional sine patterns

Manuscript received March 13, 2020.

Manuscript revised July 31, 2020.

Manuscript publicized October 7, 2020.

[†]The authors are with the Graduate School of Engineering, Osaka Prefecture University, Sakai-shi, 599-8531 Japan.

a) E-mail: iwata@cs.osakafu-u.ac.jp (Corresponding author)

b) E-mail: kise@cs.osakafu-u.ac.jp

DOI: 10.1587/transinf.2020MUP0004

Fig. 1 Overall of watermark embedding.

P^- and P^+ are defined as

$$P_{x,y}^- = \sin\left(2\pi\nu\left(\frac{x}{W} + \frac{y}{H}\right)\right) \quad (1)$$

$$P_{x,y}^+ = \sin\left(2\pi\nu\left(\frac{W-x-1}{W} + \frac{y}{H}\right)\right) \quad (2)$$

Then the shifted 2-dimensional sine patterns P^{l-} and P^{l+} are generated by horizontally shifting P^- and P^+ by l pixels, where $0 \leq l < \lambda = \max\left\{\frac{W}{\nu}, \frac{H}{\nu}\right\}$ and ν is the frequency of the 2-dimensional sine pattern relative to $W \times H$. To fix the basic 2-dimensional sine patterns, we investigated the performance of other 2-dimensional sine patterns, for example, horizontal or vertical sine patterns and slightly skewed sine patterns and found that the orthogonal 2-dimensional sine patterns employed as P^- and P^+ are the best.

The original color image I , the 2-dimensional sine patterns P^{l-} and P^{l+} are divided into $N \times N$ blocks. Moreover, every 4×2 blocks are combined as "block set". For each block set in a raster scan order, c_i is assigned to each block in a raster scan order. By doing this, one block set is corresponding to one symbol. Let (s, t) , $I^{(s,t)}$, $P^{l-(s,t)}$, $P^{l+(s,t)}$ and $C^{(s,t)}$ be the coordinate of a block, the image area of I in the (s, t) block, the image area of P^{l-} in the (s, t) block, the image area of P^{l+} in the (s, t) block and the coded bit assigned at (s, t) block, respectively. Let $F_{u,v}^{(s,t)}$, $G_{u,v}^{l-(s,t)}$, and $G_{u,v}^{l+(s,t)}$ be Discrete Fourier Transform (DFT) of $I^{(s,t)}$, $P^{l-(s,t)}$ and $P^{l+(s,t)}$, respectively. In order to decide the most suitable 2-dimensional sine pattern for embedding, $\alpha_k^{(s,t)}$ and $\beta_k^{(s,t)}$ are calculated as

$$\alpha_k^{(s,t)} = \Re\left(F_{\left[\frac{\nu}{N}\right], \left[\frac{\nu}{N}\right]}^{(s,t)} \overline{G_{\left[\frac{\nu}{N}\right], \left[\frac{\nu}{N}\right]}^{k-(s,t)}}\right) \quad (3)$$

$$\beta_k^{(s,t)} = \Re\left(F_{\left[\frac{M-\nu}{N}\right], \left[\frac{\nu}{N}\right]}^{(s,t)} \overline{G_{\left[\frac{M-\nu}{N}\right], \left[\frac{\nu}{N}\right]}^{k+(s,t)}}\right) \quad (4)$$

where M and $\Re(\cdot)$ represent the size of the blocks and the

Fig. 2 flow of detection of watermarked areas.

function to calculate the real part of a complex number, respectively. The reason why we use the DFT coefficient in Eqs. (3) and (4) is that the spectrum of the DFT coefficient is the highest in P^{l-} or P^{l+} when $(u, v) = \left(\left[\frac{\nu}{N}\right], \left[\frac{\nu}{N}\right]\right)$ or $(u, v) = \left(\left[\frac{W-\nu}{N}\right], \left[\frac{\nu}{N}\right]\right)$, respectively. Moreover, based on $\alpha_k^{(s,t)}$ and $\beta_k^{(s,t)}$, we can find the shifted 2-dimensional sine patterns which causes the largest watermarked DFT coefficient in the following process. We confirmed that the embedding with such 2-dimensional sine patterns causes robust watermarking by preliminary experiments. The watermark pattern P is obtained as

$$P^{(s,t)} = \begin{cases} P_{x,y}^{l-(s,t)} (l = \arg \max_k \alpha_k^{(s,t)}), & \text{when } C^{(s,t)} = 0 \\ P_{x,y}^{l+(s,t)} (l = \arg \max_k \beta_k^{(s,t)}), & \text{when } C^{(s,t)} = 1 \end{cases} \quad (5)$$

Then, the watermarked image I' is obtained as

$$\begin{aligned} I'_{x,y}^R &= I_{x,y}^R + a z^R P_{x,y} \\ I'_{x,y}^G &= I_{x,y}^G + a z^G P_{x,y} \\ I'_{x,y}^B &= I_{x,y}^B + a z^B P_{x,y} \end{aligned} \quad (6)$$

where a is a watermark strength and z is a color weight for embedding and extracting watermarks.

2.2 Watermark Extraction

Watermark extraction process has two processes, detection of watermarked areas, and watermark calculation. In detection of watermarked areas, the position of a watermarked area is detected from a taken picture including a watermarked area. In watermark calculation, the watermark is obtained from the detected watermarked area.

2.2.1 Detection of Watermarked Areas

The flow of detection of watermarked areas is shown in Fig. 2. Let J be a resized taken picture of size $J_W \times J_H$ pixels. The converted image $J' = (J'_{x,y})$ from J is obtained as

-1		+1
+1		-1

Fig. 3 watermark emphasis filter.

$$J'_{x,y} = z^R J^R_{x,y} + z^G J^G_{x,y} + z^B J^B_{x,y} \quad (7)$$

The watermark emphasis filter shown in Fig. 3 is applied to J' . Then the watermark pattern emphasized image $J'' = (J''_{x,y})$ is obtained as

$$J''_{x,y} = \begin{cases} 0 & (-B_2 \leq J' \leq -B_1) \\ 255 & (B_1 \leq J' \leq B_2) \\ 127 & (\text{otherwise}) \end{cases} \quad (8)$$

where B_1 and B_2 are the thresholds to judge whether pixels are watermarked or not. In watermark pattern emphasized image, 2-dimensional sine pattern appears in watermarked areas. The watermarked pixels are detected from J'' by deep learning model. Each side of the rectangle is estimated one by one by using M-estimator. The algorithm for estimating the left side is as follows: Firstly, for each row, the watermarked pixel which is firstly found by scanning from left is stored as the input of M-estimator. Note that the scanning is stopped when it reaches half of the width under the assumption that the watermarked area is roughly centered. Finally, the left side of the rectangle is estimated by using M-estimator with the stored points as the input, where we use distance function $\rho(r)$ as

$$\rho(r) = \frac{C^2}{2} \left(1 - \exp \left(- \left(\frac{r}{C} \right)^2 \right) \right) \quad (9)$$

and $C = 2.9846$. The other three sides are estimated in the same manner as described above, where the scanning directions are changed to suitable ones. Then each corner point is estimated by calculating every intersection. If estimated corner points are less than 4, there is no watermarked area (detection failure). Note that the detection success (when detection is not failure) does not mean that the watermarked area is correctly detected. The extraction of watermark is definitely incorrect when the detection of watermarked area is failed. However, the extraction of watermark is not always correct when the detection of watermarked area is succeeded. Therefore, the detection failure ratio is important to investigate the reason of incorrectly extracted watermarks.

2.2.2 Watermark Calculation

Let L be the shortest side of the detected square as the watermarked area. The projection transformation is applied to the taken picture so that the watermarked area becomes an $L \times L$ pixels square according to the detection result. The image for extraction $O = (O^R_{x,y}, O^G_{x,y}, O^B_{x,y})$ is obtained by trimming

$L \times L$ pixels watermarked area from the transformed image. The converted image $O' = (O'_{x,y})$ from O is obtained as

$$O'_{x,y} = z^R O^R_{x,y} + z^G O^G_{x,y} + z^B O^B_{x,y} \quad (10)$$

O' is divided into $N \times N$ blocks. For each block at (s, t) , $R^{(s,t)}$ is calculated as

$$R^{(s,t)} = \left| Q_{\left[\frac{M-v}{N} \right], \left[\frac{v}{N} \right]}^{(s,t)} \right| - \left| Q_{\left[\frac{v}{N} \right], \left[\frac{v}{N} \right]}^{(s,t)} \right| \quad (11)$$

where M and $Q_{u,v}^{(s,t)}$ represent the size of the blocks and the DFT coefficient obtained by applying DFT to $O'^{(s,t)}$, respectively. After calculating $R^{(s,t)}$, extracted bit $E^{(s,t)}$ is calculated as

$$E^{(s,t)} = \begin{cases} 0, & (R^{(s,t)} < 0) \\ 1, & (R^{(s,t)} \geq 0) \end{cases} \quad (12)$$

After combining every 4×2 blocks to a block set, for each block set in a raster scan order, $E^{(s,t)}$ is assigned to e_i in a raster scan order. The obtained bit sequence e is decoded by Reed Solomon coding. If the decode is succeeded, the decoded bit sequence is the watermark.

3. Datasets

Large number of images are required to train a deep learning model. For obtaining precise training data in our method, we should prepare printed watermarked images, take pictures of them and manually add the four corners of the watermarked areas of them as the ground truths. However, it takes a lot of time and effort. So we prepare two datasets, that is, a training dataset, and a fine-tuning dataset.

The training dataset is obtained from relatively large number of projection transformed non-watermarked and watermarked images. The images in this dataset are obtained automatically based on image processing. Therefore, we can prepare large number of images for this dataset without much time and effort.

The fine-tuning dataset is obtained from relatively small number of real taken pictures of non-watermarked and watermarked images. This dataset is used for the fine-tuning of the deep learning model after the training using the above training dataset including large number of images. Therefore, this dataset works well even if the number of included images is relatively small.

These datasets consist of watermark pattern emphasized images converted from projection transformed target images. Here, let the target image be an original image or a watermarked image to be captured. The following processing is a kind of data augmentation. To reproduce print-cam attacks, target images are projectively transformed within the range which can occur in actual shooting. Transformed target images are put on background images to detect watermarked areas around the borders of target images well. Here, background images of size $T_w \times T_H$ pixels are used as images of non-watermarked areas.

3.1 Training Dataset

Figure 4 shows the flow of generating the training dataset. Large number of images are used for original images and background images. As target images, S sheets of watermarked images are obtained from each original image with random S watermarks. The dataset images are obtained by applying data augmentation to these images with projection transform, where the detail of the projection transform used for this data augmentation will be described in Sect. 4.1.2. Then, the original images corresponding to the watermarked images are used as target images to obtain dataset images in the same way.

3.2 Fine-Tuning Dataset

Figure 5 shows the flow for generating a fine-tuning dataset. As the target images, taken square original images and watermarked images are used. The dataset images are obtained by applying data augmentation to these images with projection transform, where the detail of the projection transform used for this data augmentation will be described in

Fig. 4 Flow of generating training dataset.

Fig. 5 flow of making fine-tuning dataset.

Sect. 4.1.3. In this data augmentation, the U number of augmented images is generated per image. Also, dataset images are obtained by applying a watermark pattern emphasizing process directly to the taken image resized to $T_w \times T_H$ pixels. In Sect. 4.1.3, how to determine the position of target images required for clipping the target images and obtaining the ground truths for the captured images were shown.

4. Experiments

4.1 Experiments Environment

4.1.1 Devices and Deep Learning Model

HUAWEI P20 Pro was used as a mobile device. When shooting images, we used its camera with a 40 MP main RGB lens. EPSON LP-S9070 was used for printing with sRGB color correction. The resolution of the printing is 1200×1200 dpi.

We used DeepLab v3+[8] as the deep learning model for the detection of watermarked areas. This is one of the best performing models for semantic segmentation.

4.1.2 Conditions for Training Dataset

We used 986 images from PASCAL VOC 2012 [9] and 988 images from the Large Logo Dataset [10] as original images of size 400×400 pixels or more. Background images are selected from PASCAL VOC 2012, they are not selected as original images. In addition to them, a white flat image is used as a background image. The parameters for watermark embedding were $a = 2.0$, $N = 20$, $8K = 256$, $v = 60$, $z = (1.0, 0, -1.0)$. Those for watermark pattern emphasizing were $B_1 = 2$ and $B_2 = 10$. Those for generating training dataset were $S = 5$ and $T_w = T_H = 550$. The parameters of the projection transform used for the data augmentation were decided so that the four corner points of target images move based on the combination of the following transforms:

1. 85% ~ 95% image reduction
2. $-20^\circ \sim +20^\circ$ image rotation
3. move each corner point within 5% of the image length

As a result, 19,740 images were obtained. The sizes of true watermarked areas are around 400×400 pixels, where the sizes are not constant. The training dataset is divided into 15,800 training data and 3,940 validation data, where images generated from the same original image are not extended over both of them.

4.1.3 Conditions for Fine-Tuning Dataset

We used 12 images from SIDBA[†] and 6 images from IHC standard images^{††}. They are trimmed as square. The size of printed images was 12×12 cm. Each printed image was

[†]http://www.ess.ic.kanagawa-it.ac.jp/app_images_j.html

^{††}<http://www.ieice.org/iss/emm/ihc/en/image/image.php>

taken 6 times on a wall or a desk and at noon or night by HUAWEI P20 Pro. Therefore, total 24 pictures were taken for each printed image. The position of target image was determined using the finder pattern and alignment pattern used in QR codes. These detection patterns were placed far from each corner of the target image by one-sixth of the side length of the target image. As shown in Fig. 5, we cut the watermarked area of a taken picture out into a square to obtain the captured watermarked image without detection patterns. Then we put the captured watermarked image on background images with projective transformation to obtain augmented fine-tuning data. Background images were selected from PASCAL VOC 2012, where the background images were not overlapped with the images in training dataset. The parameters for watermark embedding were $a = 2.0$, $N = 20$, $8K = 256$, $v = 60$, and $z = (1.0, 0, -1.0)$. Those for watermark pattern emphasizing were $B_1 = 2$, and $B_2 = 10$. Those for generating the fine-tuning dataset were $U = 5$, and $T_w = T_H = 550$. The parameters of the projection transform used for the data augmentation were decided so that the four corner points of target images move based on the combination of the following transforms:

1. 85% ~ 90% image reduction
2. $-15^\circ \sim +15^\circ$ image rotation
3. move each corner point within 5% of the image length

The size of transformed images is 500×500 pixels. The sizes of true watermarked areas are around 400×400 pixels, where the sizes are not constant. As a result, 5,184 images were obtained.

4.1.4 Evaluation Items

We used mIoU to evaluate the performance of detection of watermarked areas to the validation data. The detection failure ratio is the ratio of the number of detection failure of the number of shots. Here, the detection failure ratio is the true negative ratio if the target image is an original image. In the case that the target image is a watermarked image, the detection failure ratio is the false negative ratio. We evaluate the performance of our deep learning model based on this ratio. The successful extraction ratio is the ratio of the number of times getting correct watermarks to the number of times shootings. The success of a watermark extraction means that the detection result is correct. So we use this to investigate whether the detection result is correct or not.

4.1.5 Environment for Test

We evaluated the performance of fine-tuning by 6-fold cross-validation. The fine-tuning dataset was divided into 6 groups in the following manner. Firstly, 3 of 18 original images described in Sect. 4.1.3 were added to each group without overlapping. Then the watermarked images and augmented fine-tuning data were added to the corresponding group to their original images. On each cross-validation, one group was used as validation data, and the

other groups were used as fine-tuning data. In the test, we used original images and watermarked images corresponding to them in the same group as those used in validation data. The parameters of watermark embedding were $a = 4.0$, $v = 60$, and $z = (1.0, 0, -1.0)$. About (N, K) , we used (12, 6) and (16, 12). The parameter (N, K) is related not only to the amount of watermark but also to the size of block used for embedding and extraction, where the block size when $(N, K) = (16, 12)$ is three quarters of that when $(N, K) = (12, 6)$. Those for watermark pattern emphasizing were $B_1 = 2$ and $B_2 = 10$. The parameter for resizing in the detection of watermarked areas was $J_W \times J_H = 550$. Tables 1–6 show the original images, the watermarked images of $(N, K) = (12, 6)$ and their PSNR, where each table is corresponding to each group.

The original images and watermarked images are printed as $15\text{cm} \times 15\text{cm}$ on A4 papers. The printed images were put on the wall. We shot them by freehand from a position about 30cm away at the horizontal angles of -20° , -10° , 0° , 10° , and 20° . Here a negative value means that images were taken from the left side. The size of taken pictures is 2160×2160 pixels trimmed from 3840×2160 pixels. The sizes of true watermarked areas in the taken pictures are around 1670×1670 pixels, where the sizes are not constant. It means that the sizes of true watermarked areas after they are resized in the detection of watermarked area are around 425×425 pixels.

4.2 Results

4.2.1 Evaluation Based on Validation Data

The mIoU after training with training dataset was 99.23% for the validation data. The mIoUs before and after fine-tuning with the fine-tuning dataset is shown in Table 7. As shown in Table 7, the mIoUs were improved by 16.58–38.65 percent points. These results confirmed that the fine-tuning worked effectively.

4.2.2 Evaluation Based on Test Data

The detection failure ratios of watermarked areas for watermarked images when $(N, K) = (12, 6)$ were 0% for all images and angles. This results confirmed that our deep learning model never overlooked the watermarked areas. The successful extraction ratios of watermarks for watermarked images when $(N, K) = (12, 6)$ are shown in Table 8. As shown in Table 8, the successful extraction ratios of watermarks for many watermarked images were 100% with several angles. Figure 6 shows an example of the results. In this case, the detected four corners of the watermarked area were correct because almost all pixels on watermarked areas were correctly detected. Figure 7 shows another result. There was a little lack of detected watermarked areas on the right side of a watermarked image. However, the watermark was extracted correctly because the detected four corners were

Table 1 group 1.

image	Aerial	Lenna	IHC1
original image
watermarked image for test
PSNR	40.8594[dB]	40.8419[dB]	40.8849[dB]

Table 2 group 2.

image	Airplane	Mandrill	IHC2
original image
watermarked image for test
PSNR	40.8522[dB]	40.8588[dB]	41.0070[dB]

Table 3 group 3.

image	Balloon	milkdrop	IHC3
original image
watermarked image for test
PSNR	40.8534[dB]	40.8529[dB]	40.9181[dB]

Table 4 group 4.

image	couple	Parrots	IHC4
original image
watermarked image for test
PSNR	40.0682[dB]	40.8968[dB]	40.8452[dB]

Table 5 group 5.

image	Earth	Pepper	IHC5
original image
watermarked image for test
PSNR	40.8715[dB]	40.9206[dB]	40.8518[dB]

Table 6 group 6.

image	Girl	Sailboat	IHC6
original image
watermarked image for test
PSNR	40.8594[dB]	40.8419[dB]	40.8669[dB]

almost correct. On the other hand, we could not extract watermarks from the following four watermarked images: couple, Girl, IHC2, and milkdrop. Figures 8–10 show the examples in which the detected watermarked areas are incorrect. In these watermarked emphasized images, there was no watermarked pattern on the left side of watermarked images. Therefore, some detection results lacked such areas as shown in Fig. 8, or include outside of the true watermarked areas as shown in Fig. 9. The extraction of watermarks was sometimes failed although the detection results were correct as shown in Fig. 10. In such cases, watermark components were broken by print-cam attacks. Table 9 shows the detection failure ratios of watermarked areas for watermarked images when $(N, K) = (16, 12)$. The detection to IHC2 and IHC6 were sometimes failed. However, half of the trials of watermark extraction were succeeded at least. Therefore,

Table 7 mIoUs before and after fine-tuning[%]. "FT" denotes whether fine-tuned or not.

FT	group 1	group 2	group 3	group 4	group 5	group 6
	79.66	57.56	71.14	62.27	68.63	55.15
✓	96.24	96.21	95.03	95.50	96.04	91.92

the positions of the watermarked areas of most watermarked images were correctly detected. These results show that the smaller block size mildly lower the performance of the detection of watermarked areas. Table 10 shows the successful extraction ratios of watermarks for watermarked images when $(N, K) = (16, 12)$. Comparing with the successful extraction ratios of watermarks for watermarked images when $(N, K) = (12, 6)$, some of them were lower. It is because the smaller size of the divided blocks caused lower robustness against print-cam attacks. Therefore, N and K should

Table 8 Successful extraction ratios of watermarks for watermarked images when $(N, K) = (12, 6)[\%]$.

group	image	-20°	-10°	0°	$+10^\circ$	$+20^\circ$
group 1	Aerial	100	100	100	100	100
	Lenna	25	0	50	50	25
	IHC1	100	100	100	100	75
group 2	Airplane	75	100	100	100	100
	Mandrill	50	100	100	100	100
	IHC2	0	0	0	0	0
group 3	Balloon	50	100	100	100	100
	milkdrop	0	0	0	0	0
	IHC3	100	100	100	100	100
group 4	couple	0	0	0	0	0
	Parrots	25	50	75	100	100
	IHC4	100	100	100	100	100
group 5	Earth	100	100	100	100	75
	Pepper	0	25	0	25	0
	IHC5	100	100	50	100	100
group 6	Girl	0	0	0	0	0
	Sailboat	100	100	100	100	100
	IHC6	0	0	50	100	25

Table 9 detection failure ratios of watermarked areas for watermarked images when $(N, K) = (16, 12)[\%]$.

group	image	-20°	-10°	0°	$+10^\circ$	$+20^\circ$
group 1	Aerial	0	0	0	0	0
	Lenna	0	0	0	0	0
	IHC1	0	0	0	0	0
group 2	Airplane	0	0	0	0	0
	Mandrill	0	0	0	0	0
	IHC2	0	0	0	0	25
group 3	Balloon	0	0	0	0	0
	milkdrop	0	0	0	0	0
	IHC3	0	0	0	0	0
group 4	couple	0	0	0	0	0
	Parrots	0	0	0	0	0
	IHC4	0	0	0	0	0
group 5	Earth	0	0	0	0	0
	Pepper	0	0	0	0	0
	IHC5	0	0	0	0	0
group 6	Girl	0	0	0	0	0
	Sailboat	0	0	0	0	0
	IHC6	50	50	25	25	0

Table 10 The successful extraction ratios of watermarks for watermarked images when $(N, K) = (16, 12)[\%]$.

group	image	-20°	-10°	0°	$+10^\circ$	$+20^\circ$
group 1	Aerial	100	100	100	100	100
	Lenna	0	0	25	25	25
	IHC1	100	100	100	100	50
group 2	Airplane	75	75	100	100	75
	Mandrill	100	100	100	100	100
	IHC2	0	0	0	0	0
group 3	Balloon	100	100	100	100	100
	milkdrop	0	0	0	0	0
	IHC3	100	100	75	100	100
group 4	couple	0	0	0	0	0
	Parrots	0	0	50	25	25
	IHC4	50	25	100	75	75
group 5	Earth	100	100	100	100	100
	Pepper	0	0	75	50	0
	IHC5	0	25	25	50	50
group 6	Girl	0	0	0	0	0
	Sailboat	75	75	75	100	100
	IHC6	0	0	0	0	0

Fig. 6 Example 1 of the detection results of watermarked areas for watermarked image when $(N, K) = (12, 6)$. **Fig. 7** Example 2 of the detection results of watermarked areas for watermarked image when $(N, K) = (12, 6)$. **Fig. 8** Example 3 of the detection results of watermarked areas for watermarked image when $(N, K) = (12, 6)$. **Fig. 9** Example 4 of the detection results of watermarked areas for watermarked image when $(N, K) = (12, 6)$. **Fig. 10** Example 5 of the detection results of watermarked areas for watermarked image when $(N, K) = (12, 6)$.

be decided under the consideration of the trade-off of the amount of watermarks and robustness. We tried to extract watermarks from watermarked logo images. However, we could not. It is because watermarks embedded in pure white areas are vulnerable to print-cam attacks due to overflowing pixel values. For example, Fig. 11 and Fig. 12 show a watermarked logo image and printed one, respectively. As shown in these figures, the watermark components in the pure white areas are perfectly removed in the printed watermarked logo

Fig. 11 Example of a watermark logo image.

Fig. 12 Example of a printed watermarked logo image.

Fig. 13 Example of a printed watermarked logo image filled with off white.

Fig. 14 The result of detection of a watermarked area in Fig. 13.

image. The replacement of pure white pixels with off white ones in original logo images can be introduced to prevent the removal of watermark components by print-cam attacks. By actually doing this, the watermarked areas of the logo image was able to be detected as shown in Fig. 13, where all RGB values of off white pixels were 246. The result of detecting watermarked areas is shown in Fig. 14.

As another challenge, we tried to extract watermarks from images which have a watermark-embedded area in part. From the results, the extraction succeeded in a few watermarked images. Figure 15 shows an example of the results. Most of the detection failure ratios were low. However, the accuracy of borders of watermarked images was not good like in Fig. 16. We should improve the accuracy of the detection of watermarked areas more for applying our method to such partially watermarked images.

The detection failure ratios of watermarked areas for original images were 100% for all images and angles. The results confirmed that the printed original images were never detected as watermarked areas. Figure 17 shows an example of the results. As shown in Fig. 17, no watermarked areas were detected although there is a texture area on a flat area in the emphasized image. Therefore the deep learning model was correctly trained so as to be able to classify whether a watermarked pixel or not based on the textures in emphasized images.

From the experiments, all of the processes took 9,511 ms on the average. Especially, the detection of watermarked areas took 8,130 ms. Therefore, our method is hard to work in real-time on current smartphones. In order to reduce processing time, using lighter deep learning models or using more high spec mobile devices can be introduced as future works.

Fig. 15 Example 1 of the detection results of watermarked areas for partial watermark embedding.

Fig. 16 Example 2 of the detection results of watermarked areas for partial watermark embedding.

Fig. 17 Example of the detection results of watermarked areas for an original image.

5. Conclusion

In this paper, we proposed a novel digital watermarking method using deep learning for detecting watermarked areas. Our method is able to extract a watermark without adding markers to watermarked images nor registering original images. The experiments confirmed that our method can extract 96 bits watermarks with freehand shooting a printed watermarked image by smartphones.

Future works include reducing the detection time of watermarked areas using a deep learning model and improving our method so that watermarks can be extracted more stably from various images.

References

- [1] D.W. INCORPORATED, "QR code.com," <https://www.qrcode.com/en/>.
- [2] T. Nakamura, A. Katayama, M. Yamamuro, and N. Sonehara, "Fast watermark detection scheme from analog image for camera-equipped cellular phone," *The transactions of the Institute of Electronics, Information and Communication Engineers, D-II*, vol. 87, no. 12, pp. 2145–2155, 2004.
- [3] A. Pramila, A. Keskinarkaus, V. Takala, and T. Seppänen, "Extracting watermarks from printouts captured with wide angles using computational photography," *Multimedia Tools and Applications*, vol. 76, no. 15, pp. 16063–16084, 2017.
- [4] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Increasing the capturing angle in print-cam robust watermarking," *J. Systems and Software*, vol. 135, pp. 205–215, 2018.

- [5] K. Yamanaka, M. Iwata, and K. Kise, "A consideration of digital watermarking method to printed images," in Abstracts of the 2013 Symposium on Cryptography and Information Security, p.96, Jan. 2013.
- [6] H. Ogawa, M. Iwata, and K. Kise, "Digital watermarking method using image retrieval to printed images without finder patterns and alignment pattern," Proc. 2016 Symposium on Cryptography and Information Security (SCIS2016), Jan. 2016.
- [7] X. Zhong and F.Y. Shih, "A robust image watermarking system based on deep neural networks," arXiv preprint arXiv:1908.11331, 2019.
- [8] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," ECCV, pp.833–851, 2018.
- [9] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [10] A. Sage, E. Agustsson, R. Timofte, and L. Van Gool, "Lld - large logo dataset - version 0.1," <https://data.vision.ee.ethz.ch/cvl/lld>, 2017.

Koichi Kise received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, Osaka, Japan, in 1986, 1988, and 1991, respectively. From 2000 to 2001, he was a Visiting Professor with the German Research Center for Artificial Intelligence (DFKI), Germany. He is currently a Professor with the Department of Computer Science and Intelligent Systems and the Director of the Institute of Document Analysis and Knowledge Science (IDAKS), Osaka Prefecture University, Japan. His major research activities are in analysis, recognition and retrieval of documents, images and activities. Prof. Kise is also a member of IEEE, ACM, IPSJ, IEEJ, ANLP, and HIS. He was a recipient of the Best Paper Award of IEICE, in 2006, the IAPR/ICDAR Best Paper Awards, in 2007 and 2013, the IAPR Nakano Award, in 2010, the ICFHR Best Paper Award, in 2010, and the ACPR Best Paper Award, in 2011. He worked as the Chair of the IAPR Technical Committee 11 (reading systems), and a member of the IAPR conferences and meetings committee. He is currently the Editor-In-Chief of the international journal of document analysis and recognition.

Hiroyuki Imagawa received the B.S. degree in Engineering from Osaka Prefecture University in 2018. His research interests include digital watermarking to printed.

Motoi Iwata received M.E. and D.E. degrees in computer and systems sciences from Osaka Prefecture University, in 1999 and 2005, respectively. From 1999 to 2016, he was an Assistant Professor in the Graduate School of Engineering, Osaka Prefecture University. Since 2016, he has been an Associate Professor in the Graduate School of Engineering, Osaka Prefecture University. His current research focuses on digital watermarking, data hiding, image retrieval, comics image analysis and reading-life

log. He is a member of the Imaging Society of Japan, the Institute of Image Information and Television Engineers, and IEEE.