

Collusion Resistant Watermarking for Deep Learning Models Protection

Sayoko Kakikura*, Hyunho Kang**, Keiichi Iwamura*

**Department of Electrical Engineering, Tokyo University of Science, Japan*

***Department of Electronic Engineering, National Institute of Technology, Tokyo College, Japan*

kakikura_sayoko@sec.ee.kagu.tus.ac.jp, kang@tokyo-ct.ac.jp, iwamura@ee.kagu.tus.ac.jp

Abstract— Deep learning has been used in many fields, such as image classification and data analysis. Training a high-performance model is expensive; thus, its property value is high. Watermarking is a representative technology providing intellectual property protection for models. In this study, we proposed white box watermarking using a modified Barni's method (our previous study) for image watermarking. Our method is applicable to pre-trained models because the watermark is embedded in the parameters of the network without training. Additionally, the proposed method embeds multiple watermarking into neural networks using different keys. We evaluated the method using two different networks: 5-layer convolutional neural networks trained on MNIST and ResNet-50 trained on CIFAR-10 datasets. The experimental results show that our proposed approach can embed 10 watermarks with less than 0.1% loss of accuracy, and it detects them completely even after 90% of the parameters are pruned.

Keywords— Copyright protection, Intellectual property protection, Multi-layer neural network, Watermarking, White box watermarking, Collusion resistant watermarking

I. INTRODUCTION

Deep learning has become an indispensable technology for a wide range of fields in recent years. Because developing a high-quality deep learning model requires significant resources, such as large amounts of training data, machine resources, and a significant amount of time, the models have significant value as intellectual property, that must be protected.

Watermarking is a widely used technology for protecting the copyright of digital content, such as image and audio data. Neural network watermarking was first proposed by Uchida et al. [1] and since then, several studies have been published. Uchida et al. specified the following requirements for neural network watermarking:

- Fidelity: The watermark does not degrade the performance of the host network task
- Robustness: The watermark is robust against model modifications, for example, model compression
- Efficiency: The watermark embedding and extraction processes at high speed

In this study, we aim to fulfill the aforementioned three requirements and embed multiple watermarks without collisions. Our method applies our previous watermarking

method [2], which extends Barni's method [3] into watermarking for convolutional neural networks (CNN). The image watermarking developed in our previous study is robust to noise and can embed multiple watermarks independently. The ability to embed multiple watermarks makes it easy to add more watermarks as more people take ownership of the model. In addition, watermark embedding does not require a training process. Our method is applicable for pre-trained networks and watermarks can be easily edited by the owner of the model.

II. RELATED WORKS

Watermarking techniques for neural networks are classified based on the accessible information at the time of watermark extraction: white box and black box paradigm. The white box paradigm provides access to all parameters in a host network, allowing information to be embedded into the parameters [1], [4]. Because black box paradigm can only access the output of the final layer, some researchers have proposed training networks to deliberately produce incorrect output for a specific input, and use it as a watermark [5]–[7].

The method proposed by Uchida et al. involves embedding a watermark into network parameters using a cost function with a parameter regularizer. The watermarks are embedded while training a host network.

Fan et al. [4] proposed a passport-based approach in which the host network is appended with a passport layer. This passport layer also functions as a watermark. When a fake passport layer is appended instead of a valid passport layer, the performance of the host network degrades considerably.

III. PROPOSED METHOD

Our proposed watermark embedding and detecting algorithm is based on [2], which modifies Barni's method to archive a higher correlation.

A. Watermark Embedding

1) **Target Parameters Determination:** Select parameter tensor T to embed the watermark in the host network (e.g., weights in convolution layer, fully connected layer).

2) **Parameter Pre-processing:** Flatten T and trim it to a length of N^2 . N^2 is observed to be less than 80% of the number

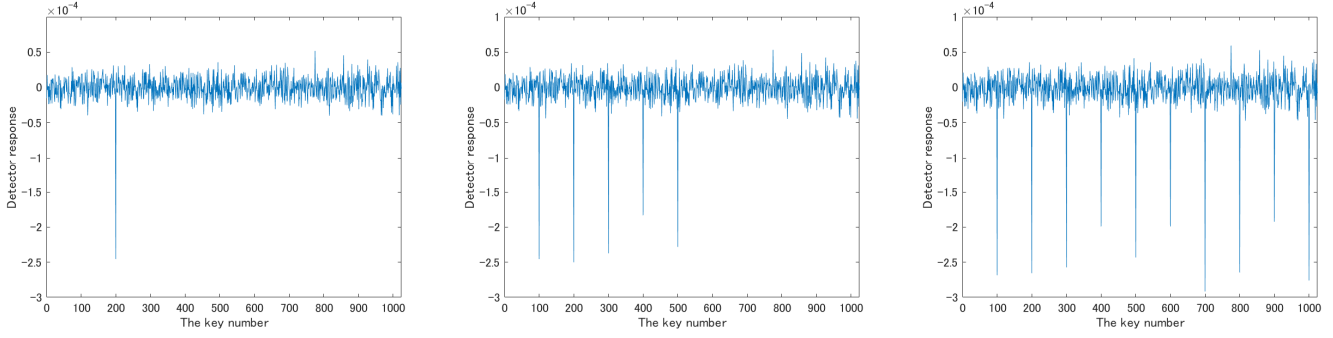


Figure 1. Watermark detection on 5-layers CNN (Left: 1 watermark embedded, Center: 5 watermarks, Right: 10 watermarks)

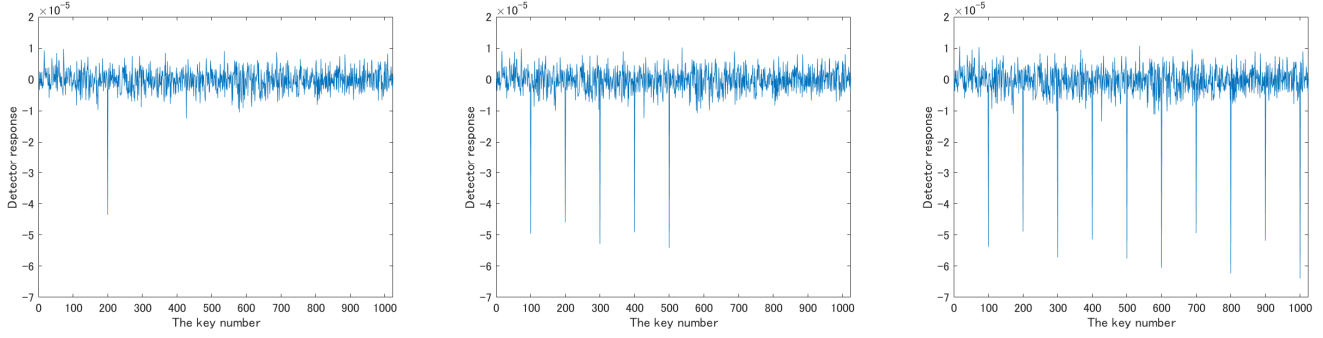


Figure 2. Watermark detection on ResNet-50 (Left: 1 watermark embedded, Center: 5 watermarks, Right: 10 watermarks)

of elements in T . The trimmed T is reshaped to $N \times N$, which is T_{sq} .

3) Watermark Generation: Let an arbitrary positive integer k be the key. The elements of T_{sq} are shuffled based on k , and then the discrete cosine transformed (DCT). The generated DCT coefficient matrix is denoted as E and the mid-frequency domain of E is denoted as E_a . E_a consists of M coefficients. Watermark is embedded using the following equation:

$$W = E_a - P_{mark}$$

where P_{mark} represents a sequence of uniformly distributed pseudo-random numbers in the range of the minimum and maximum values of E_a , with k as the seed value. E_{a+w} is defined as the following equation:

$$E_{a+w} = E_a + \alpha \cdot W$$

where α represents the embedding strength.

4) Embedding: Replace the mid-frequency domain of E with E_{a+w} , perform inverse DCT, and reorder by k (denoted as T_{sq+w}). Finally, put T_{sq+w} back into T and replace the host network parameters.

To embed two or more watermarks, repeat step 3 while changing k .

B. Watermark Detecting

The mid-frequency domain of DCT coefficient matrix E'_{a+w} is extracted from the watermark embedded network using the same processes used for watermark embedding. P'_{mark} is calculated using x as the seed value ($x = 1, 2, \dots, n$ >

k) and the minimum and maximum values of E'_{a+w} , a range of uniformly distributed pseudo-random numbers.

The correlation value between E'_{a+w} and P'_{mark} is determined by the following equation:

$$corr = \frac{1}{M} \sum E'_{a+w} \cdot P'_{mark}$$

where M represents the length of E'_{a+w} and P'_{mark} , in other words, the size of the embedded area. If an attack on the watermark is not considered, then

$$E'_{a+w} = E_{a+w} = (1 + \alpha) E_a - P_{mark}, \quad P_{mark} \cong P'_{mark}$$

and there is a strong correlation between E'_{a+w} and P'_{mark} in the case of $x = k$. Therefore, k is detected by searching the x values that show a strong correlation. Even if an attack adds noise to E'_{a+w} , k can be detected if the correlation value when $x = k$ is sufficiently greater than the correlation value when $x \neq k$.

IV. EXPERIMENTAL RESULTS

We evaluated our proposed method with two different neural networks for classification tasks: a 5-layers CNN described in Table 1 trained on MNIST, and ResNet-50 pre-trained on ImageNet and retrained on CIFAR-10. ResNet-50 [8] is a CNN, which is 50 layers deep; we used the pre-trained version released by MathWorks as the base of transfer learning.

A. Experimental Settings

1) Target Parameters: On the 5-layer CNN, the watermark was embedded into 100×2000 weight parameters between the pooling and hidden layers. On ResNet-50, the

target parameters are the weights in the last convolution layer. The size of the weights is $1 \times 1 \times 512 \times 2048$.

TABLE 1. CNN ARCHITECTURE FOR MNIST

Layer	Remark	Activation Function
Input	28×28 nodes	-
Convolution	20 conv. filters (9×9)	ReLU
Pooling	1 mean pooling (2×2)	-
Hidden	100 nodes	ReLU
Output	10 nodes	Softmax

TABLE 2. PERFORMANCES BEFORE-AFTER WATERMARK EMBEDDING

	Num of Watermarks	Accuracy	
		5-layer CNN (MNIST)	ResNet-50 (CIFAR-10)
Not embedded	-	0.9465	0.8456
Embedded	1	0.9470	0.8456
	5	0.9475	0.8466
	10	0.9475	0.8453

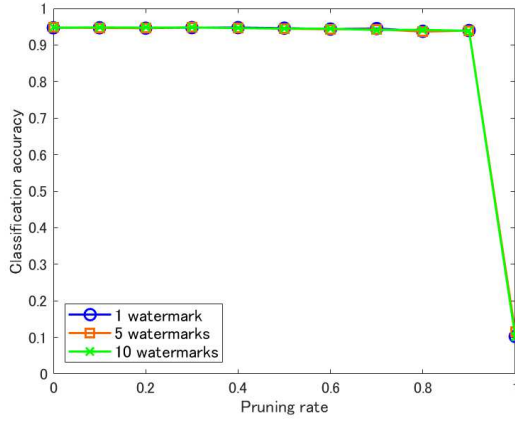


Figure 3. Pruning attack on the 5-layers CNN embedded with different number of watermarks

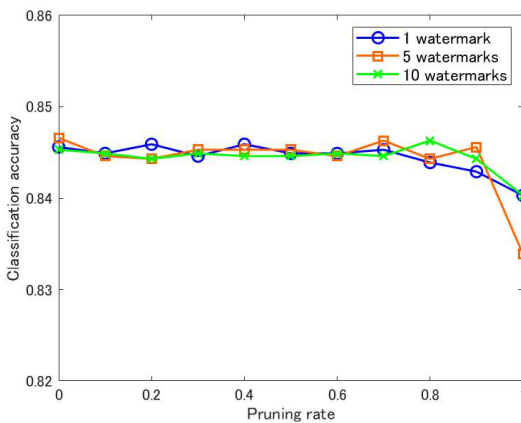


Figure 4. Pruning attack on ResNet-50 embedded with different number of watermarks

2) Embedding Scheme Variables: We set $N = 400$, $\alpha = 0.04$, $M = 24000$, and $n = 1024$. The key values were set to $k = 200$ for embedding one watermark, $k = 100i$ ($i = 1, 2, \dots, 5$) for embedding five watermarks, and $k = 100i$ ($i = 1, 2, \dots, 9, 10$) for embedding ten watermarks.

3) Test Datasets: We prepared 2,000 images from MNIST for 5-layer CNN, 3,000 images from CIFAR-10 for ResNet-50 as the evaluating networks.

B. Embedding Results

We evaluated the proposed method from two perspectives:

- Classification accuracy of the host model: Whether the embedding of the watermark affects the accuracy of the original task
- Watermark detection accuracy: Whether the embedded watermark can be detected correctly

As shown in Table 2, there is no loss of accuracy after watermark embedding in either network. The number of watermarks embedded in the weights has little effect on accuracy. These results confirmed that our watermark has little effect on the classification task.

The results of the correlation detection are shown in Figures 1 and 2. In both the 5-layers CNN and ResNet-50, only the correct keys show strong correlations. Even when multiple watermarks were embedded, the key number corresponding to each watermark was detected independently.

C. Robustness against Model Compression

Model compression is one of the attacks against neural network watermarking because it changes the parameters of the network and thus affects the embedded watermark. In this study, we evaluated the robustness of the watermark against model compression, particularly against weight pruning. In pruning, $p\%$ of the watermarked weights with low absolute values were set to zero. p (or p divided by 100) is known as the pruning rate.

Figures 3 and 4 show the classification accuracy for different pruning rates. As can be seen in Figure 3, the 5-layer CNN maintains nearly the same accuracy from no pruning to 90% pruning; however, it sharply decreases when 100% weights are cut off. ResNet-50 also shows a decrease in classification accuracy when pruning 100% weights.

Figures 5 and 6 show the detector responses from the networks that prune 90% of the weights watermark embedded. Even if almost watermarked parameters are cut off, all embedded watermarks are detected.

V. CONCLUSIONS

In this study, we proposed a watermark embedding method for CNNs based on the image watermarking method. The results show that our method has minimal impact on image classification task and robustness to pruning attack. Regardless of the number of watermarking embedded, our method can detect all watermarking completely.

On ResNet-50, when the weights were 100% pruned, the watermark was completely removed; however, the

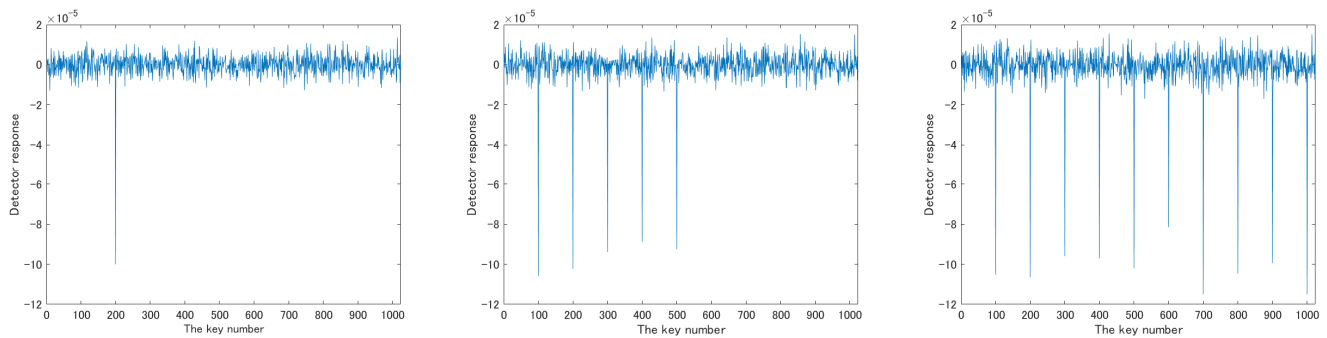


Figure 5. Watermark detection on 90% pruning 5-layers CNN (Left: 1 watermark embedded, Center: 5 watermarks, Right: 10 watermarks)

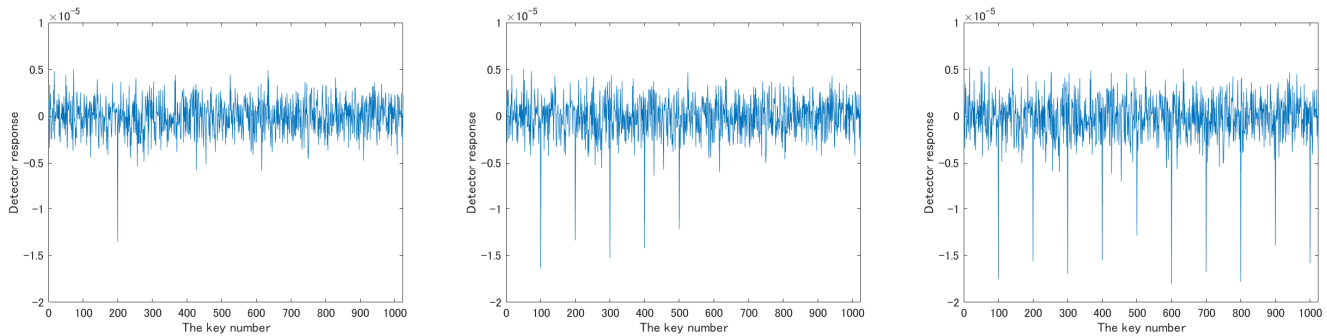


Figure 6. Watermark detection on 90% pruning ResNet-50 (Left: 1 watermark embedded, Center: 5 watermarks, Right: 10 watermarks)

classification accuracy was barely degraded. There is considerable improvement in the selection of the layer for embedding the watermark in complex CNNs.

REFERENCES

- [1] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proc. ICMR '17*, 2017, pp. 269-277.
- [2] H. Kang, and K. Iwamura, "Collusion-resistant watermarking using modified Barni method," *Ubiquitous Information Technologies and Applications*, Lecture Notes in Electrical Engineering, Berlin, Germany: Springer, 2014, vol. 280, pp.627-634.
- [3] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "Robust watermarking of still images for copyright protection," in *Proc. 13th ICOSP*, 1997, pp. 499-502.
- [4] L. Fan, K. W. Ng, and C. S. Chan, "Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks," in *Proc. NIPS'19*, 2019, pp. 4714-4723.
- [5] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *Proc. ASIACCS'18*, 2018, pp. 159-172.
- [6] J. Guo, and M. Potkonjak, "Watermarking deep neural networks for embedded systems," in *Proc. ICCAD'18*, 2018, pp. 1-8.
- [7] E. L. Merrer, P. Pérez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Computing and Applications*, vol. 32, Jul. 2020, pp. 9233-9244.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR'16*, 2016, pp. 770-778.

Sayoko Kakikura received her Bachelor of Engineering (B.E), in field of electrical engineering, from Tokyo University of Science, Japan, in 2021. She is currently pursuing the M.E degree with Tokyo University of Science, Japan. Her main research interests include watermarking and deep learning.

Hyunho Kang is currently an Associate Professor in the Department of Electronic Engineering at National Institute of Technology, Tokyo College, Japan; he has held this position since April 2017. He received his Ph.D. from the University of Electro-Communications, Tokyo, in 2008. From 2008 to August 2010, he was a Researcher/Assistant Professor at Chuo University, Tokyo, where he was part of a team that developed Biometric Security technologies. From September 2010 to March 2013, he was an AIST Postdoctoral Researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Japan, where his research work focused mainly on the evaluation of physical unclonable functions. From April 2013 to March 2017, he was an Assistant Professor in the Department of Electrical Engineering at Tokyo University of Science, Japan. His main interests are machine learning, deep learning, information security applications, multimedia security (steganography, digital watermarking), biometric security and physical unclonable functions. Dr. Kang is a senior member of Institute of Electronics, Information and Communication Engineers (IEICE) and a member of Information Processing Society of Japan (IPSI).

Keiichi Iwamura received B.S. and M.S. degrees in Information Engineering from Kyushu University in 1980 and 1982, respectively. During 1982-2006, he was with Canon Inc. He received a Ph.D. from Tokyo University. He is now a Professor at the Tokyo University of Science. His subjects are coding theory, information security, and digital watermarking. Dr. Iwamura is a fellow of IEICE and a fellow of IPSJ.