DeepWatermark: Embedding Watermark into DNN Model

Minoru Kuribayashi†, Takuro Tanaka, Nobuo Funabiki
Okayama University, Okayama, Japan
E-mail: † kminoru@okayama-u.ac.jp

***Abstract*—For the protection of trained deep neural network(DNN) model, it has been studied to embed a watermark into the weights of DNN. However, the amount of changes in the weights is large in the conventional methods. In addition, it is reported that the presence of hidden watermark can be detected from the analysis of weight variance, and that the watermark can be modified by effectively adding noise to the weight. In this paper, we focus on the fully-connected layers and apply a quantization-based watermarking method to the weights sampled from the layers. The advantage of the proposed method is that the changes caused by embedding watermark is much smaller and measurable. This is effective against the problems of previous works. The validity of the proposed method is quantitatively evaluated by changing the conditions during the training of DNN model. The results include the impact of training for DNN model, effective embedding method, and high robustness.**

I. Introduction

Due to the rapid progress in computer performance, deep learning techniques have been actively investigated in various research fields. The core techniques in the deep learning are the architecture of deep neural network (DNN) and its trained model using big data. To promote the research on deep learning, the inheritance of common architecture plays an important role both in academic and in industrial researches. On the other hand, training the model designed by using the common architecture has different property. As a large amount of computing resources and a huge amount of data are required to train such a model, the protection against unauthorized copying of the trained model becomes an important issue. In particular, the weight values among nodes of neural network retain the great importance on the performance of DNN-based system.

In previous works, as a countermeasure against illegal copying of trained DNN model, the introduction of watermarking technique into the DNN-based system has been studied in [1], [2]. In the method, a vector of watermark information is embedded into the vector of weights selected from a specified layer of convolutional neural network (CNN). Using a secret matrix, the vector of weights is updated at each training process so that the multiplication with the matrix becomes close to the watermark vector. The efficiency of the embedding watermark is dependent on the selection of the matrix, and the penalty induced by the watermark embedding operation is quantitatively evaluated in the experiments. The operation is performed in parallel with the training of model by using supervised dataset. In terms of loss function, a binary cross entropy is used to control the degradation of the performance of CNN-based system. In addition to the original loss function, the embedding loss function is considered in the training so that the weights are updated according to the watermark as well as supervised dataset. In [3], [4], the selection of feature vector is refined from the method in [1], [2]. However, a vulnerability is reported in [5]. After embedding watermark, the variance of weight values becomes large, which leaks information about a hidden message in a certain CNN-layer. It is also revealed that the watermark is easily modified by overwriting. The other interesting approach is based on a visible watermarking method presented in [6]. Similar to the case of adversarial examples, the watermarked image is classified to a different label due to the embedded visible watermark. The disadvantage of the method is the low robustness against the same watermark removal attacks that have been investigated for the last few decades.

In this paper, we propose a novel watermarking approach for DNN model under the assumption that the accuracy of DNN model is not sensitive to the difference of local minima. It is known [7], [8] that in DNN models there are many local minima that are likely to yield an accuracy very close to another. Based on the generic tendency in a DNN model, the proposed method modifies some weights selected from a DNN model without considering their impact on the accuracy. Instead, the amount of changes caused by embedding watermark is controlled to be small by using the idea of quantization index modulation (QIM) method [9]. As the changes caused by embedding watermark are also converged with the progress of training process, the embedding loss is ignored in the proposed method. In addition, the techniques of random permutation and dither modulation [10], [11], which method is called DM-QIM, are employed to the enhancement of the secrecy of watermark. As such operations are managed by a secret key, it is difficult for attackers to find the presence of watermark in a given DNN model. The intentional modification of embedded watermark is also difficult without the secret key, and high robustness against addition of noise can be achieved by controlling the quantization step size as well as the number of weights selected from a DNN model. Therefore, the proposed method retains higher secrecy and robustness than the conventional works.

We quantitatively evaluated the effects of embedding watermark on the training process by changing step size of quantization process in the DM-QIM method. At the training

process of DNN model, a group of supervised dataset is given and the loss measurement is calculated to update the weights among nodes in neural networks. For each epoch of such a process, the sensitivities to the training efficiency are evaluated in our simulation. When the step size is increased, the amount of change in weights caused by the embedding watermark becomes large, and thus it requires more epochs for the training of DNN model to be converged. Our experimental results revealed that the changes are so rapidly converged that it is sufficient to perform the embedding operation only at the first epoch in the training process. It is also confirmed that the proposed method retains high robustness against attacks such as overwriting and pruning less important weights.

II. Related Works

A. Watermarking for DNN Model

Depending on the attacker's point of view, watermarking techniques for neural networks can be classified into black-box and white-box methods. A watermark can be extracted by only giving queries to a target model in a black-box method, while a whitebox watermark needs to get parameters in the model in order to extract the watermark.

The first white-box method was developed by Uchida et al. [1], [2]. Rouhani et al. [3] presented an improved version and their research group proposed its application of fingerprinting for tracing illegal users. For a given DNN model, a bit-string of watermark information is embedded into the parameters of one or more network layers. Considering the degradation of performance, the above conventional methods avoid directly modifying the parameters to embed watermark. In order not to impair the performance of an original DNN model in its original task, a binary cross entropy for regularizing the watermark embedding task is introduced in the cost function in training process.

Watermark information is denoted by a vector $\boldsymbol{w}$ of length k. Let $X \in \mathbb{R}^{k\times n}$ be a matrix to be kept secret, and $\boldsymbol{p}$ be vector of weights in network layers to be watermarked which length is n. Then, the binary cross entropy $H(\boldsymbol{p})$ is defined by

$$H(\boldsymbol{p}) = -\sum_{i=1}^{k} \big(w_i \log(y_i) + (1-w_i)\log(1-y_i)\big), \quad (1)$$

where

$$y_i = \sigma\big(\sum_{j=0}^{n} X_{ij} p_j\big) \quad (2)$$

and $\sigma()$ is an activation function like a sigmoid function. Each watermark bit w_i is embedded so that the following equation becomes true.

$$w_i = \begin{cases} 1 & y_i \geq 0.5 \\ 0 & y_i < 0.5 \end{cases} \quad (3)$$

The embedding matrix X is regarded as a secret key, and it must be carefully generated so that the distribution of watermarked weights becomes unnatural. Wang et al. [5] pointed out the problem that the significant difference in the distribution can be observed from watermarked weights, and presented a method to remove the watermark by an overwriting attack. The main reason of the problem comes from the fact that the above method modifies the weights considerably in order to satisfy the condition given by Eq.(3).

B. Quantization-Based Watermarking

A watermarking algorithm takes a host vector selected from a given content, watermark, and a secret key as a input, and outputs a watermarked vector. Among some watermarking methods, the quantization index modulation (QIM) method [9] is prove to be optimal with respect to the amount of distortions caused by embedding watermark.

Let x be an element in the host vector, and let $w \in \{0,1\}$ be a watermark. In the QIM method, x is quantized into the nearest even/odd point from quantization grids determined by a quantization step δ. The selection of even and odd point is determined by w. For example, suppose that $x = 123$ and $\delta = 10$. If $w = 0$, then the quantized value is 120; otherwise 130. It is noticed that the quantized value becomes a member of the set $\{0, \pm 10, \pm 20, \pm 30, \cdots\}$. If a malicious party knows the embedding algorithm, each element of the host vector can be found by simply observing the values. Therefore, the DM operation is introduced to enhance secrecy.

In the DM operation, a random number r is identically and independently selected from the range $[-\delta/2, \delta/2]$. Then, a watermark w is embedded into x using r as follows:

$$\overline{x} = \text{DM-QIM}(x, w, \delta, r) \quad (4)$$

$$= \begin{cases} \delta \cdot \lfloor \frac{x+r}{\delta} \rfloor - r & \text{if } \lfloor \frac{x+r}{\delta} \rfloor \bmod 2 = w; \\ \delta \cdot \big(\lfloor \frac{x+r}{\delta} \rfloor + 1\big) - r & \text{otherwise.} \end{cases} \quad (5)$$

The above operation quantizes $x+r$ into the nearest even/odd value according to the watermark bit w, and then subtract r from the quantized value. At the receiver's end, the watermark w is extracted as follows.

$$w = \Big\lfloor \frac{x^\star + r + \frac{\delta}{2}}{\delta} \Big\rfloor \bmod 2. \quad (6)$$

Notice that $\overline{x}$ is not a multiple of δ. As the watermark bit w cannot be extracted without r, it is regarded as a secret key in this method.

III. Proposed Method

In the conventional methods, the loss function for embedding watermark as well as DNN model were used to insert the watermark information during the training process. Different from the approach, in this research, we propose a digital watermarking method that has less effect on model parameters for DNN by applying the DM-QIM method [10][11].

We assume that the amount of change caused by embedding watermark converges with the progress of training process. At the first embedding operation, the amount of change is expected to be maximum. After the first embedding, the weights of DNN model is updated in the training process. It slightly modifies the weights, and hence, it can be regarded as the addition of tiny noise to the watermarked weights. Thus,

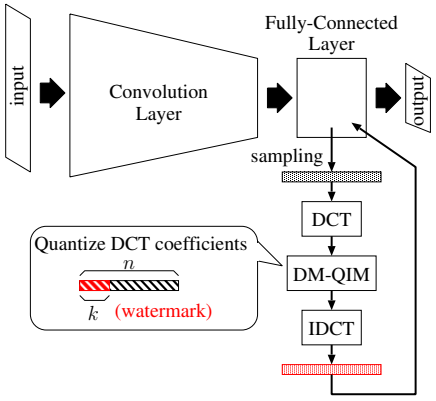

Fig. 1. Schematic diagram of proposed watermarking method.

the change at the second embedding is relatively smaller than that of first one. After some epochs of training process, the change is expected to be negligibly small. It means that the loss function for embedding watermark is unnecessary in the above process.

In [9], the amount of changes caused by the embedding process can be estimated by a statistical analysis, and it is proven to be minimum. Hence, the impact on the accuracy of DNN model is expected to be smaller than the other watermarking methods. In order to make the presence of watermark information hard to find, the watermark information should not be directly embedded in the weights. According to a secret key, n weights are randomly selected and their frequency components are calculated by discrete cosine transform (DCT). Among n frequency components, we choose k DCT coefficients for embedding so that the changes with respect to the selected weights are diffused. Furthermore, before and after the quantization process in the embedding watermark, the dither modulation controlled by the secret key is performed to improve the confidentiality.

A. Embedding Process

Suppose that watermark is embedded into a deep learning system based on convolutional neural network (CNN). In convolution layers, the weights in a fixed window size are updated to find better local filtering operations, while the connections among nodes are fitted in fully-connected(FC) layers. To share the common architecture of CNN, the convolution layers are imported from known CNN-based system, and by using the technique of fine-tuning the FC layers are adjusted for a given instance of applications. As shown in the Fig. 1, weights at FC layers are targeted for embedding watermark in which a fine-tuning is performed on a trained model. The description of the procedure is explained below.

1) Select n weights from FC layers according to a secret key key, which is denoted by a vector $\boldsymbol{f}$:
$$\boldsymbol{f} = (f_0, f_1, \ldots, f_{n-1}). \tag{7}$$
2) Perform DCT to the vector $\boldsymbol{f}$, and obtain the frequency components $\boldsymbol{F}$.
3) For each bit of watermark $\boldsymbol{w}$:
$$\boldsymbol{w} = (w_0, w_1, \ldots, w_{k-1}), \tag{8}$$
the corresponding frequency components F_i modified by using the DM-QIM method.
$$\overline{F}_i = \text{DM-QIM}(F_i, w_i, \delta, r_i), \tag{9}$$
where δ is the quantization step and r_i is the random dither signal generated by using a pseudo-random number generator PRNG and a secret key key.
$$\boldsymbol{r} = \text{PRNG}(key) = (r_0, r_1, \ldots, r_{k-1}) \tag{10}$$
4) Perform the inverse DCT to the vector $\overline{\boldsymbol{F}}$, and replace f_i with the watermarked weight $\overline{f}_i$ in the FC layers.

By using the above method, embedded signal as watermark is spread over the while sampled weights. Due to the characterisitcs of QIM method, the distortions caused by the embedding are small. In addition, the introduction of secret key, the analysis of the presence of hidden message becomes difficult from the observation of weights in the FC layers.

B. Amount of Changes in Weights

The amount of changes in the weights $\boldsymbol{f}$ selected for embedding can be measured. As each element in $\boldsymbol{f}$ are randomly sampled from FC layers according to a secret key, the frequency components $\boldsymbol{F}$ can be regarded as random variables which energy E_f can be calculated from the L2 norm of the vector $\boldsymbol{f}$.
$$E_f = \sum_{i=0}^{n} f_i^2 \tag{11}$$

For a given step δ, F_i for $i = 0, 1, \ldots k-1$ are modified by using the DM-QIM method. As the value of F_i is regarded as a random number, the amount of changes after embedding watermark is varying in the range $[-\delta, \delta]$. Then, the expected energy ϵ of changes for each F_i can be calculated by the following formula.
$$\epsilon = \frac{1}{2\delta}\int_{-\delta}^{\delta} t^2 dt = \frac{\delta^2}{3} \tag{12}$$

Hence, the total energy E_w of k-bit watermark can be given as follows:
$$E_w = \frac{k\delta^2}{3}. \tag{13}$$

After the inverse DCT, the energy is spread over n elements of $\boldsymbol{f}$. Thus, the expected watermark energy at each weight is E_w/n.

The average change amount ϵ derived by Eq.(12) is valid only at the first embedding process. As weights in the neural network are gradually updated in a training process, the watermark signal may be distorted at each epoch of the training process. It means that the noise induced during the training process is added to the watermarked DCT coefficients $\overline{F}_i$ in Eq.(9). It is noted that the amount of noise at each weight is small and considered to converge to 0 with the progress of training the DNN model.

C. Robustness

There is a trade-off among capacity of watermark, distortion, and robustness against noise. When the number of weights selected from the FC layers is n, the maximum embeddable information amount is n bits. Under the constraint of a fixed distortion level, the step size can be increased when $k < n$. Suppose that the number of selected weights is $n = 2048$ and the step size is $\delta = 10$. When 2048-bit watermark is embedded into the weights, the total change amount is $E_w = 204800/3$. If the amount of watermark information is $k = 128$ bits, then we can choose the step size $\delta = 40$ under the same distortion level because the total change amount is calculated by Eq.(13). Since the watermark information is quantized and embedded in the frequency component of the weight vector, its change spreads over the entire weight vector and its influence is diffused.

From an attacker's point of view, it is difficult to add noise only to the k DCT coefficients without the secret key because of the lack of information about the selection of weights from a huge number of candidates in the FC layers as well as their order in the selected ones. Even if some weights in $\overline{\boldsymbol{f}}$ are happened to be modified by adding noise, the noise energy is spread over all DCT coefficients $\overline{\boldsymbol{F}}$. With the increase of noise, the performance of DNN model must be degraded, and hence, it is difficult for an attacker to add strong noise to remove the watermark. From the above characteristics, we can say that the proposed method retains high robustness against noise.

D. Impact on DNN Model

During a training process, the weights in a DNN model are modified so that the output of the loss function becomes smaller. Depending on the selection of supervised dataset and initial setup, the result of training is different. It is well-known that deep neural networks have many local minima, and that all local minima are likely to be very close to a global minimum [7], [8]. Thus, there are some local minima where the loss function becomes small, and moving to another local solution does not make much difference in the performance.

With the increase of energy of watermark signal, it must take more time for training the DNN model to be converged. As discussed in Section III-B, the proposed method employs a variant of QIM watermarking method in which the amount of distortions caused by embedding watermark is small and measurable. Embedding watermark results in the move to one of closest local minima from the initial minimum, which makes the analysis of watermark difficult. In addition, We can control the energy of watermark by setting watermark bit-length k and the quantization step size δ.

IV. Experiments

In this section, we conducted experiments to the proposed method by implementing on a DNN model. The performance is evaluated in terms of accuracy, transition of loss value in a training process, and convergence speed. Since the training process begins with a scratch, the results may be fluctuated depending on the initial setup. Therefore, in this experiment, we tried 10 times and calculated their average.

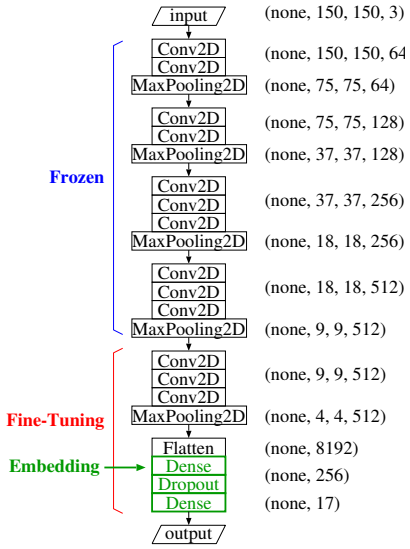

Fig. 2. Network architecture in the experiment.

A. Experimental conditions

In this study, a DNN model developed by Visual Geometry Group of Oxford University called VGG16 [12] is used as a trained model. This model was trained using over 1 million images from the ImageNet [13] database. Based on this trained model, we use a fine-tuning method by implanting new FC layers after the final convolutional layer of the VGG16 model.

Figure 2 illustrates the network architecture of our fine-tuning model based on the VGG16. The number of nodes at the final convolutional layer is 8192 (=$4 \times 4 \times 512$) in the VGG16, and these node are connected to new FC layers with 256 nodes. The number of candidates for selecting weights from the FC layers is more than 2 millions ($8192 \times 256 + 256 \times 17$). Among such a huge candidates, we select only $n = 2048$ weights according to a secret key.

As the dataset for training the fine-tuning model, we use 17 Category Flower Dataset [14] provided by Visual Geometry Group of Oxford University. The training is performed by dividing each of the 17 classes into 50 pieces of training data, 10 pieces of validation data, and 20 pieces of test data.

B. Convergence of Embedding Operation

We investigate the effects of embedding watermark by using the proposed method, and evaluate the convergence tendency of the watermarked model. It is assumed that the embedding operation is performed every epoch after training a DNN model so that the distortions of weights caused by embedding watermark convergent to be negligibly small.

At first, we measure the energy of distortions $\overline{E}$ in the n

TABLE I
COMPARISON OF THE THEORETICAL VALUE E_w AND EXPERIMENTAL VALUE $\overline{E}$ AT THE FIRST EPOCH.

	step size δ			
	2	4	8	16
experimental	169.81	680.10	2717.36	10870.40
theoretical	170.67	682.67	2730.67	10922.67

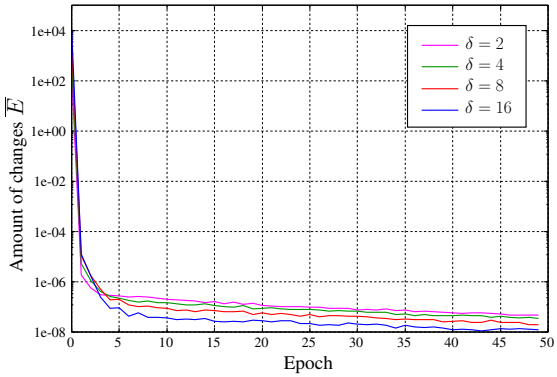

Fig. 3. Convergence of $\overline{E}$ with the progress of training process.

TABLE II
DEGRADATION OF TEST ACCURACY WITH RESPECT TO STEP SIZE δ, WHERE THE ACCURACY AND LOSS OF ORIGINAL TRAINED MODEL IS 0.8859 AND 0.2860, RESPECTIVELY.

	step size δ			
	2	4	8	16
accuracy	0.8894	0.8838	0.8750	0.8741
loss	0.3463	0.3823	0.4673	0.4490

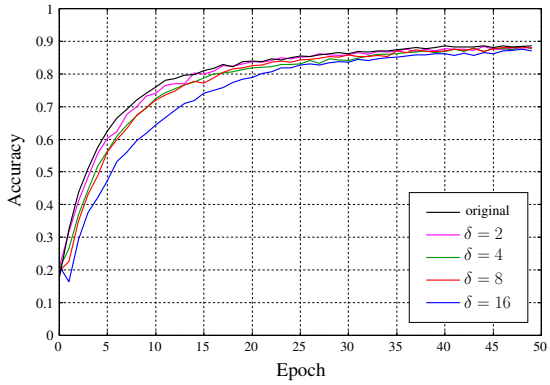

Fig. 4. Transition of validation accuracy with the progress of training process.

weights selected from FC layers.

$$\overline{E} = \sum_{i=1}^{n} (\overline{f}_i - f_i)^2 \tag{14}$$

As discussed in Section III-B, the expectation of $\overline{E}$ is given by Eq.(13) at the first epoch. Table I shows the experimental values for varying the step size δ at the first epoch when the bit-length of watermark information is $k = 128$. It confirms the validity of the theoretical analysis because the theoretical values are almost coincident with the experimental values.

Next, the convergence speed of $\overline{E}$ is evaluated with the progress of training process. In order to investigate the difference in the effect of the step size, the changes in $\overline{E}$ are observed with respect to δ={2, 4, 8, 16}. The results depicted in Fig. 3 show that $\overline{E}$ converges to 0 at a fairly early stage. The training DNN model at each epoch gives considerably less impacts on the embedded watermark at the early stage of the training process. In other words, the watermark is less sensitive to the update of weights at each epoch in this experimental condition.

C. Performance of DNN Model

Due to the injection of watermark signal into the weights in a DNN model, the accuracy must be degraded from the original trained model. We evaluate the performance of watermarked model in terms of accuracy and loss measurements. Table II enumerates the degradation of test accuracy. It is observed that the accuracy is gradually decreased with the step size δ though the amount of degradation is small.

The results of checking the progress of training are shown in Fig. 4 and Fig. 5. It is observed from Fig. 4 that the accuracy at the early stage of training process becomes lower with the increase of step size δ. Nevertheless, the final accuracy is approaching to the same level to the original model. It can be said that the embedding watermark sacrifices the training efficiency because it requires more epochs to be converged.

Even if the embedding operation is performed through all epochs, the changes on the weights are appeared only at the first epoch. After the first epoch, the effect of embedding operation seems negligibly small with respect to the accuracy and loss. Therefore, we measure the performance of the simplified method such that the embedding operation is performed only at the first epoch.

The solid lines in Fig. 6 and Fig. 7 show the results when the step size is $\delta = 16$. Comparing with the cases where the embedding process is performed through all epochs and only at the first epoch, no remarkable difference can be observed from these figures. From this result, it can be said that it is

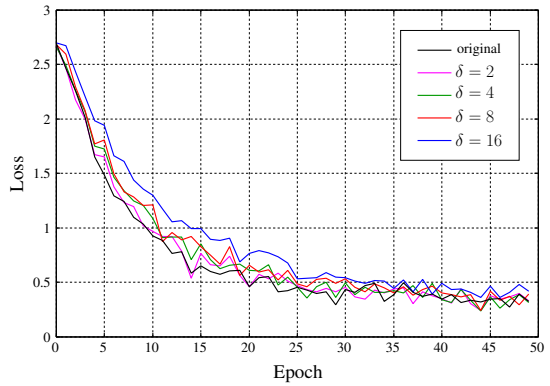

Fig. 5. Transition of validation loss with the progress of training process.

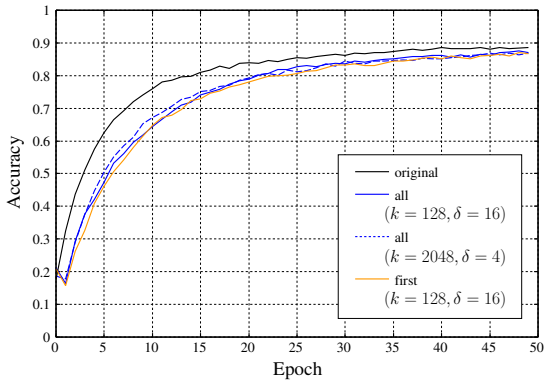

Fig. 6. Comparison of validation accuracy when the embedding operation is performed through all epochs and only at the first epoch.

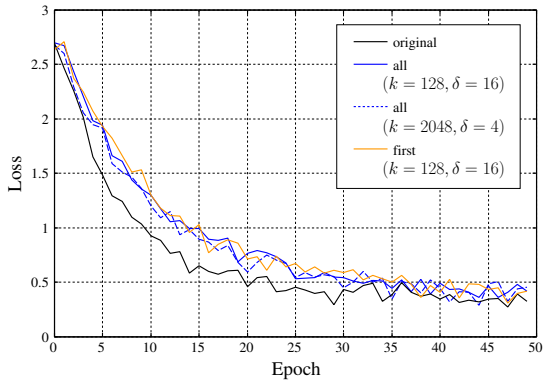

Fig. 7. Comparison of validation loss when the embedding operation is performed through all epochs and only at the first epoch.

enough to perform the embedding operation only at the first epoch. These figures also show the results of the case when $k = 2048$ and $\delta = 4$, where the expectation of total watermark energy E_w is same as the case of $k = 128$ and $\delta = 4$ as discussed in Section III-B. As the results in accuracy and loss are very close with each other, it confirms the validity of the statistical analysis.

D. Secrecy and Robustness

The amount of changes on the weights is small in the DM-QIM method, and the watermark signal embedded in $k = 128$ DCT coefficients spread over n weights. Thus, the increase of variance of the watermarked weights is small and controllable by setting a proper step size δ. In addition, we choose $n = 2048$ weights selected from more than 2 million candidates according to a secret key in the proposed method. Without the secret key, it is extremely difficult to analyze such a small increase of variance in the tiny subset of parameters.

We conduct an experiment about the overwriting attack such that a different watermark is embedded in a watermarked DNN model by using a different key. If the same step size δ is used, the noise observed at the detection of original watermark is negligibly small in the experiment, and hence, we can extract

TABLE III
DEGRADATION OF ACCURACY WHEN AN OVERWRITING ATTACK IS PERFORMED. A WATERMARKED MODEL IS CREATED BY EMBEDDING WATERMARK ONLY AT THE FIRST EPOCH WITH $\delta = 16$ AND ITS TEST ACCURACY AND LOSS IS 0.8718 AND 0.4457, RESPECTIVELY.

	step size δ			
	2	4	8	16
accuracy	0.5371	0.5053	0.5156	0.5191
loss	2.6307	3.3209	3.7377	3.0912

the original watermark with no error by using its corresponding secret key. When different information is embedded in some weights in a watermarked model, most of the selected weights are different from the watermarked ones because of the different secret key. Therefore, it is difficult to interfere the original watermark signals by the overwriting attack. On the other hand, due to the addition of different watermark signal, the accuracy of the DNN model is remarkably dropped from the original model. Table III shows the degradation of test accuracy, where an original watermark is embedded by using $\delta = 16$ and the other watermark is embedded to the watermarked model without any additional training process. Due to the significant changes in the selected weights, the trained model is greatly affected by the overwriting attack. If an attacker tries to overwrite watermark in a trained model, it requires the same training process to obtain the comparative performance. Therefore, retraining is indispensable to remove watermark from a watermarked DNN model. We also evaluate the robustness against a pruning attack such that the connection with some relatively less important weights in a DNN model are deleted. In case of $\delta = 16$, $k = 128$, and $n = 2048$ in our experimental condition, no detection error occurred for the pruning rate less than 50%.

V. CONCLUSIONS

In this paper, we proposed a deep watermarking method that has little influence on the DNN model performance. The embedding operation is performed on the frequency components of the sampled weights so that the embedded watermark information can be diffused over the samples. Since a secret key is used to the selection of the weights as well as the embedding watermark in the DM-QIM method. the secrecy and robustness of the watermark are assured in the proposed method. One of the advantage of the method is the small distortion level, and it can be controlled by setting a quantization step size. Instead of embedding loss function, we perform the proposed embedding operation after the epochs in the training process of DNN model. It is revealed from our experiments that it is sufficient to perform the embedding operation only at the first epoch in our experimental condition. The detailed analysis of the effects for different DNN models and embedding parameters are still left for our future works.

ACKNOWLEDGMENT

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

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] Y. Nagai, Y. Uchida, S. Sakazawa, and S. Satoh, "Digital watermarking for deep neural networks," *International Journal of Multimedia Information Retrieval*, vol. 7, pp. 3–16, 2018.

[3] B. D. Rouhani, H. Chen, and F. Koushanfar, "DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks," in *Proc. ASPLOS'19*, 2019, pp. 485–497.

[4] H. Chen, B. D. Rouhani, C. Fu, J. Zhao, and F. Koushanfar, "DeepMarks: A secure fingerprinting framework for digital rights management of deep learning models," in *Proc. ICMR'19*, 2019, pp. 105–113.

[5] T. Wang and F. Kerschbaum, "Attacks on digital watermarks for deep neural networks," in *Proc. ICASSP'19*, 2019, pp. 2622–2626.

[6] J. Zhang, Z. Gu, J. Jang, H. Wu, M. Ph. Stoecklin, H. Huang, and I. Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *Proc. ASIACCS'18*, 2018, pp. 159–172.

[7] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, *The loss surfaces of multilayer networks*, Artificial Intelligence and Statistics, 2015.

[8] Y. N. Dauphin, R. Pascanu, Ç. Gülçehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Proc. NIPS'14*, 2014, pp. 2933–2941.

[9] B. Chen and G. Q. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[10] M. Kuribayashi, T. Fukushima, and N. Funabiki, "Data hiding for text document in PDF file," in *Proc. IIHMSP'17*, 2017, pp. 390–398.

[11] M. Kuribayashi, T. Fukushima, and N. Funabiki, "Robust and secure data hiding for PDF text document," *IEICE Trans. Information and Systems*, vol. E102-D, no. 1, pp. 41–47, 2019.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR'15*, 2015.

[13] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR'09*, 2009, pp. 248–255.

[14] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. CVPR'06*, 2006, vol. 2, pp. 1447–1454.