

# Original Paper

# Efficient Fine-Tuning with Domain Adaptation for Privacy-Preserving Vision Transformer

Teru Nagamori<sup>1</sup>, Sayaka Shiota<sup>1</sup> and Hitoshi Kiya<sup>1\*</sup>

<sup>1</sup>*Tokyo Metropolitan University, 6-6, Asahigaoka, Hino, Tokyo, Japan*

---

## ABSTRACT

We propose a novel method for privacy-preserving deep neural networks (DNNs) with the Vision Transformer (ViT). The method allows us not only to train models and test with visually protected images but to also avoid the performance degradation caused from the use of encrypted images, whereas conventional methods cannot avoid the influence of image encryption. A domain adaptation method is used to efficiently fine-tune ViT with encrypted images. In experiments, the method is demonstrated to outperform conventional methods in an image classification task on the CIFAR-10 and ImageNet datasets in terms of classification accuracy.

*Keywords:* privacy-preserving, domain adaptation, image classification, vision transformer

## 1 Introduction

Deep neural networks (DNNs) have been deployed in many applications including security-critical ones such as biometric authentication and medical image

---

\*Corresponding author: Hitoshi Kiya, [kiya@tmu.ac.jp](mailto:kiya@tmu.ac.jp). This study was partially supported by JSPS KAKENHI (Grant Number JP21H01327) and JST CREST (Grant Number JPMJCR20D3).

analysis. In addition, training a deep learning model requires a huge amount of data and fast computing resources, so cloud environments are increasingly used in various applications of DNN models. However, since cloud providers are not always reliable in general, privacy-preserving deep learning has become an urgent problem [8, 4, 23].

One of the privacy-preserving solutions for DNNs is to use encrypted images to protect visual information in images for training and testing models [7, 14]. In this approach, images are transformed by using a learnable encryption method, and encrypted images are used as training and testing data. The use of models trained with encrypted images enables us to use state-of-the-art learning algorithms without any modification. However, all conventional learnable encryption methods [20, 19, 13, 1, 16, 17] have a problem, that is, the performance of encrypted models degrades compared with models without encryption.

Traditional cryptographic methods such as homomorphic encryption [15, 22] are one of the other privacy-preserving approaches, but the computation and memory costs are expensive, and it is not easy to apply these methods to state-of-the-art DNNs directly. Federated learning [23, 11] allows users to train a global model without centralizing the training data on one machine, but it cannot protect privacy during inference for test data when a model is deployed in an untrusted cloud server.

Accordingly, we focus on the problem of degraded model performance that occurs when models are trained with encrypted images. In this paper, we consider reducing the performance degradation of encrypted models when using encrypted images in the Vision Transformer (ViT) [5], which has high performance in image classification, and we propose a domain adaptation method to reduce the influence of encryption. In experiments, the proposed method is demonstrated to maintain almost the same performance as models trained with plain images in terms of image classification accuracy even when using encrypted images.

The rest of this paper is structured as follows. Section 2 presents related work on image encryption for deep learning and the Vision Transformer. Regarding the proposed method, Section 3 includes an overview, image encryption, and fine-tuning with domain adaptation. Experiments for verifying the effectiveness of the method, including classification performance, training efficiency, and comparison with conventional methods, are presented in Section 4, and Section 5 concludes this paper.

## 2 Related work

### 2.1 Image Encryption for Deep Learning

Various image transformation methods with a secret key, often referred to as perceptual image encryption or image cryptography, have been studied so

far for many applications [8]. In this paper, we focus on learnable images transformed with a secret key, which have been studied for deep learning. Learnable encryption enables us to directly apply encrypted data to a model as training and testing data. Encrypted images have no visual information on plain images in general, so privacy-preserving learning can be carried out by using visually protected images. In addition, the use of a secret key allows us to embed unique features controlled with the key into images. Adversarial defenses [2] and access control [9] are carried out with encrypted data using unique features.

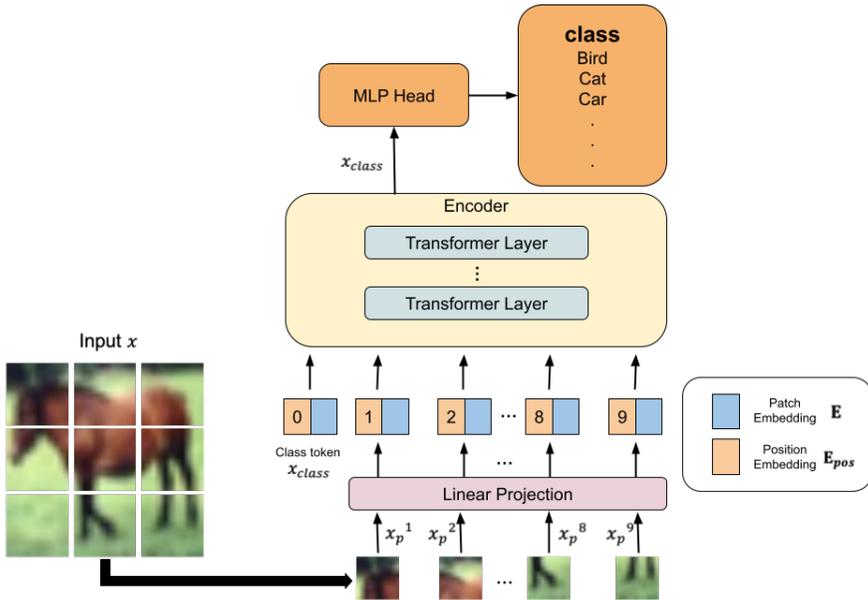
Tanaka first introduced a block-wise learnable image encryption method (LE) with an adaptation layer [20], which is used prior to a classifier to reduce the influence of image encryption. Another encryption method is a pixel-wise encryption (PE) method in which negative-positive transformation (NP) and color component shuffling are applied without using any adaptation layer [19]. However, both encryption methods are not robust enough against ciphertext-only attacks, as reported in [3, 7]. To enhance the security of encryption, LE was extended to an extended learnable image encryption method (ELE) by adding a block scrambling (permutation) step and a pixel encryption operation with multiple keys [13]. However, ELE still has an inferior accuracy compared with using plain images, even when an additional adaptation network (denoted as ELE-AdaptNet hereinafter) is applied to reduce the influence of the encryption [13]. Moreover, large-size images cannot be applied to ELE because of the high computation cost of ELE-AdaptNet.

Recently, block-wise encryption was also pointed out to have a high similarity with isotropic networks such as ViT and ConvMixer [10], and the similarity enables us to reduce performance degradation [1, 16, 10]. In addition, neural network-based image encryption methods without any block-wise encryption, called NeuraCrypt [24] and Color-NeuraCrypt [17], have been proposed, but these methods also have the same performance degradation problem as conventional block-wise encryption methods.

Accordingly, we propose a domain adaptation method so that the performance degradation problem is solved by combining block-wise encryption with domain adaptation under the use of ViT.

## 2.2 Vision Transformer

The Vision Transformer (ViT) [5] is commonly used in image classification tasks and is known to provide a high classification performance. As shown in Figure 1, in ViT, an input image  $x \in \mathbb{R}^{h \times w \times c}$  is divided into  $N$  patches with a size of  $p \times p$ , where  $h$ ,  $w$ , and  $c$  are the height, width, and number of channels of the image. Also,  $N$  is given as  $hw/p^2$ . Afterward, each patch is flattened into  $x_p^i = [x_p^i(1), x_p^i(2), \dots, x_p^i(L)]$ . Finally, a sequence of embedded patches is given as

Figure 1: Architecture of Vision Transformer ( $N=9$ )

$$z_0 = [x_{class}; x_p^1 \mathbf{E}; x_p^2 \mathbf{E}; \dots; x_p^i \mathbf{E}; \dots; x_p^N \mathbf{E}] + \mathbf{E}_{pos}, \quad (1)$$

where

$$\begin{aligned} \mathbf{E}_{pos} &= ((e_{pos}^0)(e_{pos}^1) \dots (e_{pos}^i) \dots (e_{pos}^N)), \\ L &= p^2 c, x_{class} \in \mathbb{R}^D, x_p^i \in \mathbb{R}^L, e_{pos}^i \in \mathbb{R}^D, \\ \mathbf{E} &\in \mathbb{R}^{L \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}. \end{aligned}$$

$x_{class}$  is a class token,  $\mathbf{E}$  is an embedding (patch embedding) that linearly maps each patch to dimensions  $D$ ,  $\mathbf{E}_{pos}$  is an embedding (position embedding) that gives position information to patches in the image,  $e_{pos}^0$  is the position information of a class token,  $e_{pos}^i$  is the position information of each patch, and  $z_0$  is a sequence of embedded patches. Afterward,  $z_0$  is input into the transformer encoder. The encoder outputs only the class token, which is a vector of condensed information on the entire image, and it is used for classification.

Previous studies have indicated that when DNN models are trained with encrypted images, the performance of the models is degraded compared with models trained with plain images. In contrast, it has been pointed out that block-wise encryption has similarity with models that have embedding structures such as ViT [1, 10]. Accordingly, we focus on the relationship between

block-wise image encryption and the embedding structure of ViT to avoid the influence of image encryption even when encrypted images are used for training models.

### 3 Proposed Method

#### 3.1 Overview

Figure 2 shows the procedure for fine-tuning a model by using the proposed domain adaptation. Since a model pre-trained with plain images as in [13] is prepared in advance, the fine-tuning with encrypted images that we will discuss is highlighted in the figure.

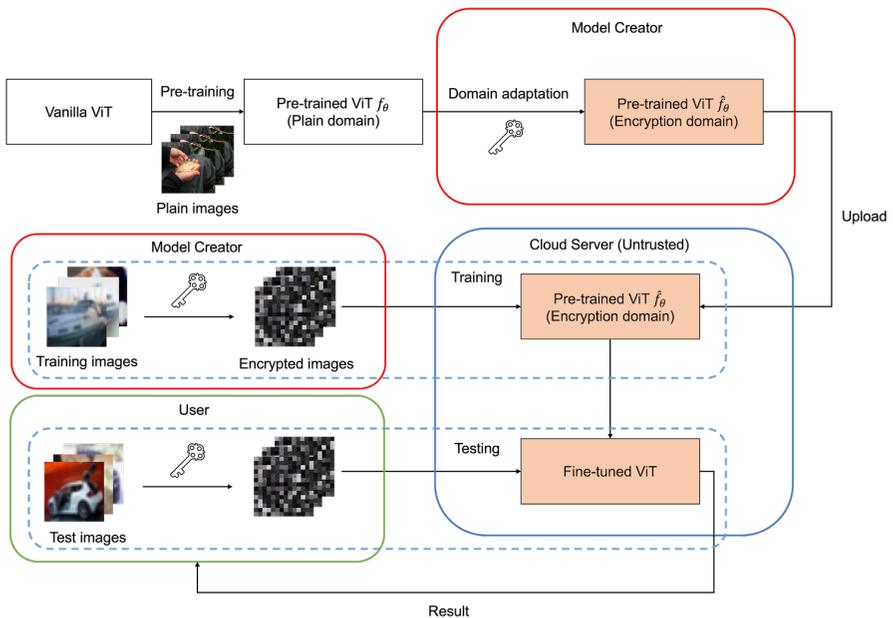


Figure 2: Overview of proposed fine-tuning

A model creator applies the domain adaptation to the pre-trained model  $f_\theta$  to generate a model  $\hat{f}_\theta$  with a key. Model  $\hat{f}_\theta$  is then uploaded to a cloud service such as Amazon Web Services (AWS), which may be untrusted. Model  $\hat{f}_\theta$  is fine-tuned by using images encrypted with the key. In this paper, we use a block-wise encryption method, which consists of block scrambling and pixel shuffling, to generate encrypted images as an example of block-wise encryption [8]. In block scrambling, an image is divided into non-overlapped

blocks (patches), and the divided blocks are randomly permuted with a key. In contrast, pixel shuffling is used to randomly permute the position of pixels in each block as described later.

Finally, test images are encrypted with the same keys as those used for fine-tuning by a user, and the encrypted test images are input to the fine-tuned model.

The proposed method is adaptable to ViT-based models with the position and patch embeddings, because the method is carried out by using the embeddings. Some other models such as ConvMixer [21] with only patch embedding can be fine-tuned by slightly modifying the proposed method. However, the method is not applicable to models without any embedding.

### 3.2 Image Encryption

Model  $f_\theta$  is fine-tuned by using encrypted images as shown in Figure 2. Figure 3 shows the encryption procedure used in this paper. Details on the procedure are given below.

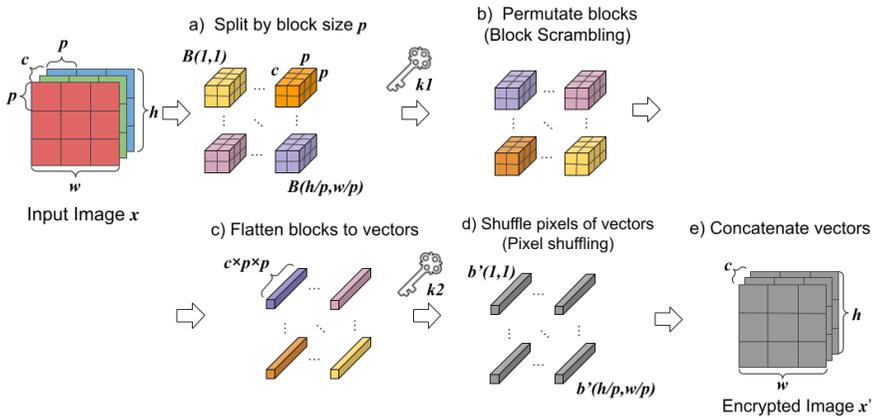


Figure 3: Procedure of image encryption

(a) Divide an image  $x \in \mathbb{R}^{h \times w \times c}$  into non-overlapped blocks with a size of  $p \times p$  such that  $B = (\{B_1, \dots, B_i, \dots, B_N\})^\top$ ,  $B_i \in \mathbb{R}^{p^2 \times c}$ .

(b) Permute blocks with secret key  $k_1$  such as

$$\begin{aligned} \hat{B} &= \mathbf{E}_{\text{bs}} B \\ &= (\{\hat{B}_1, \dots, \hat{B}_i, \dots, \hat{B}_N\})^\top, \hat{B}_i \in \mathbb{R}^{p^2 \times c} \end{aligned} \quad (2)$$

where  $\hat{B}_i \in \{B_1, \dots, B_N\}$ , and a permutation matrix  $\mathbf{E}_{\mathbf{bs}}$  is generated with key  $k_1$  as follows.

- 1) Using key  $k_1$ , generate a random integer vector with a length of  $N$  as

$$l_e = [l_e(1), l_e(2), \dots, l_e(i), \dots, l_e(N)], \quad (3)$$

where

$$\begin{aligned} l_e(i) &\in \{1, 2, \dots, N\}, \\ l_e(i) &\neq l_e(j) \text{ if } i \neq j, \\ i, j &\in \{1, \dots, N\}. \end{aligned}$$

- 2)  $k_{(i,j)}$  is given as

$$k_{(i,j)} = \begin{cases} 0 & (j \neq l_e(i)) \\ 1 & (j = l_e(i)) \end{cases}. \quad (4)$$

- 3) Define  $\mathbf{E}_{\mathbf{bs}} \in \mathbb{R}^{N \times N}$  as

$$\mathbf{E}_{\mathbf{bs}} = \begin{bmatrix} k_{(1,1)} & k_{(1,2)} & \dots & k_{(1,N)} \\ k_{(2,1)} & k_{(2,2)} & \dots & k_{(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ k_{(N,1)} & k_{(N,2)} & \dots & k_{(N,N)} \end{bmatrix}, \quad (5)$$

For example,  $\mathbf{E}_{\mathbf{bs}}$  is given by, for  $N = 3$ ,

$$\mathbf{E}_{\mathbf{bs}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (6)$$

- (c) Flatten each block  $\hat{B}_i$  into a vector  $\hat{b}_i$  with a length of  $L = p \times p \times c$  such that

$$\hat{b}_i = [\hat{b}_i(1), \dots, \hat{b}_i(L)]^\top, \quad (7)$$

where  $\hat{b}_i$  is the same as  $(x_p^i)^\top$  in Equation 1.

- (d) Shuffle pixels in  $\hat{b}_i$  with secret key  $k_2$  such as

$$\begin{aligned} b'_i &= (\mathbf{E}_{ps} \hat{b}_i)^\top \\ &= [b'_i(1), \dots, b'_i(L)], \end{aligned} \quad (8)$$

where  $\mathbf{E}_{ps}$  is a matrix generated by the same procedure as  $\mathbf{E}_{\mathbf{bs}}$  ( $N$  is replaced with  $L$ ).

Also, Equation 8 is expressed as

$$b'_i = (\mathbf{E}_{ps} \hat{b}_i)^\top = (\mathbf{E}_{ps} (x_p^i)^\top)^\top = x_p'^i. \quad (9)$$

- (e) Concatenate the encrypted vectors  $b'_i (i \in \{1, \dots, N\})$  into an encrypted image  $x'$ .

Figure 4 shows an example of images encrypted with this procedure, where block size  $p$  was 16. In the proposed method, a common key is constantly applied to all training and test images.

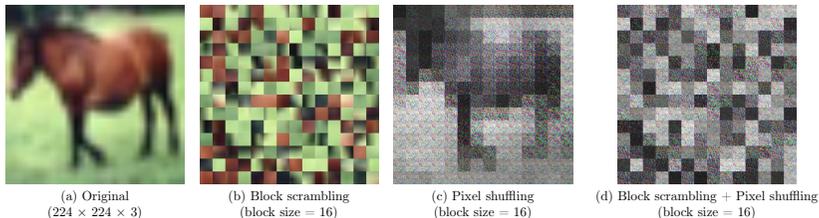


Figure 4: Example of encrypted images

### 3.3 Fine-tuning with Domain Adaptation

As shown in Figure 2, model  $f_\theta$  pre-trained in the plain domain is transformed into model  $\hat{f}_\theta$  to reduce the influence of encryption prior to the fine-tuning of  $f_\theta$  by using the proposed domain adaptation. The procedure of the adaptation is summarized here.

The domain adaptation is carried out in accordance with the embedding structure of ViT.  $\mathbf{E}_{\text{bs}}$  in Equation 5 is used to permutate blocks in an image, so it has a close relationship with position embedding  $\mathbf{E}_{\text{pos}}$  in Equation 1, where  $\mathbf{E}_{\text{pos}}$  includes the position information of a class token  $x_{\text{class}}$ . In contrast,  $\mathbf{E}_{\text{bs}}$  does not consider the information of  $x_{\text{class}}$ . To fill the gap between  $\mathbf{E}_{\text{pos}}$  and  $\mathbf{E}_{\text{bs}}$ ,  $\mathbf{E}_{\text{bs}}$  is extended as

$$\mathbf{E}'_{\text{bs}} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & k_{(1,1)} & k_{(1,2)} & \dots & k_{(1,N)} \\ 0 & k_{(2,1)} & k_{(2,2)} & \dots & k_{(2,N)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & k_{(N,1)} & k_{(N,2)} & \dots & k_{(N,N)} \end{bmatrix}, \quad (10)$$

$$\mathbf{E}'_{\text{bs}} \in \mathbb{R}^{(N+1) \times (N+1)}.$$

In the domain adaptation,  $\mathbf{E}_{\text{pos}}$  is transformed as in

$$\hat{\mathbf{E}}_{\text{pos}} = \mathbf{E}'_{\text{bs}} \mathbf{E}_{\text{pos}}. \quad (11)$$

By using Equation 11,  $\mathbf{E}_{\text{pos}}$  can be adapted to images permuted by block scrambling. Similarly,  $\mathbf{E}$  is adapted to pixels that are randomly replaced by pixel shuffling, as in

$$\hat{\mathbf{E}} = \mathbf{E}_{\text{ps}}\mathbf{E}. \quad (12)$$

From Equations 2, 9, 11, and 12, under the use of image encryption and domain adaptation, an adapted sequence of embedded patches in Equation 1 is replaced with

$$\begin{aligned} \hat{z}_0 &= \mathbf{E}'_{\text{bs}}[x_{\text{class}}; (\mathbf{E}_{\text{ps}}(x_p^1)^\top)^\top \mathbf{E}_{\text{ps}}\mathbf{E}; \dots (\mathbf{E}_{\text{ps}}(x_p^N)^\top)^\top \mathbf{E}_{\text{ps}}\mathbf{E}] + \mathbf{E}'_{\text{bs}}\mathbf{E}_{\text{pos}} \quad (13) \\ &= \mathbf{E}'_{\text{bs}}[x_{\text{class}}; x_p^1 \hat{\mathbf{E}}; \dots x_p^N \hat{\mathbf{E}}] + \mathbf{E}'_{\text{bs}}\mathbf{E}_{\text{pos}}. \end{aligned}$$

In accordance with Equations 11 and 12,  $f_\theta$  with  $\mathbf{E}_{\text{pos}}$  and  $\mathbf{E}$  is transformed to  $\hat{f}_\theta$  with  $\hat{\mathbf{E}}_{\text{pos}}$  and  $\hat{\mathbf{E}}$  as shown in Figure 2, where the difference between  $f_\theta$  and  $\hat{f}_\theta$  is only a sequence of embedding patches  $\hat{z}_0$  in Equation 13.

Finally, test images are applied to  $\hat{f}_\theta$  in the adaptation domain. In our framework, as shown Figure 5, test images are input to a model after fine-tuning  $\hat{f}_\theta$  with encrypted images. In addition, test images are required to be encrypted by using the same keys as those used for fine-tuning the model.

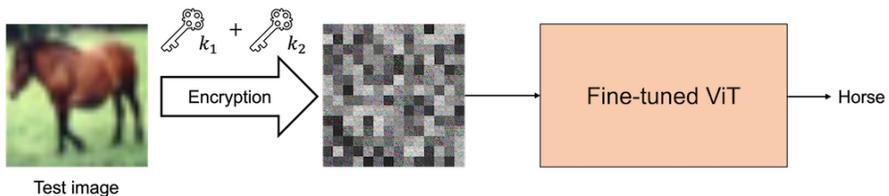


Figure 5: Test procedure

## 4 Experiments

The effectiveness of the proposed method is verified in terms of the accuracy of image classification and the efficiency of model training.

### 4.1 Setup

We conducted image classification experiments on the CIFAR-10 [12] and CIFAR-100 datasets [12]. CIFAR-10 consists of 60,000 images with 10 classes

(6000 images for each class), and CIFAR-100 consists of 60,000 images with 100 classes (600 images for each class), where 50,000 images were used for training, and 10,000 images were used for testing. In addition, we used the Imagenette dataset [6], which is a subset of the ImageNet dataset with 10 classes. We also fine-tuned a ViT-B\_16 model pre-trained with the ImageNet21k dataset, so we resized the image size from  $32 \times 32 \times 3$  to  $224 \times 224 \times 3$ . Training was carried out under the use of a batch size of 32, a learning rate of 0.001, a momentum of 0.9, and a weight decay of 0.0005 using the stochastic gradient descent (SGD) algorithm for 15 epochs. A cross-entropy loss function was used as the loss function. The block size of the encryption was set to 16 to match the ViT patch size.

#### 4.2 Classification Performance

Table 1 shows the classification accuracy of models trained with encrypted images. When training models without the proposed domain adaptation, the accuracy decreased compared with that of using plain images. In particular, when applying the combined use of pixel shuffling and block scrambling, the classification accuracy significantly decreased due to the influence of encryption. In contrast, when training models with the domain adaptation, the accuracy was almost the same as that of using plain images (Baseline). Accordingly, we confirmed that the proposed domain adaptation was effective in fine-tuning pre-trained models with encrypted images.

Table 1: Classification accuracy (%) with and without proposed method (CIFAR-10)

Encryption	Domain Adaptation	
	Yes	No
Pixel shuffling	<b>99.05</b>	97.78
Block scrambling	<b>98.93</b>	94.25
Pixel shuffling + Block scrambling	<b>98.98</b>	72.86
Plain (Baseline)	99.00	

#### 4.3 Training Efficiency

When training a model, it is important to consider not only the classification performance but also the training efficiency. Training efficiency means the time required for model training and loss convergence. The use of encrypted images should be avoided as it significantly increases the time required for training and loss convergence compared with models trained using plain images.

Figure 6 shows the classification accuracy and loss value of three models per epoch during training and testing where the number of epochs was increased

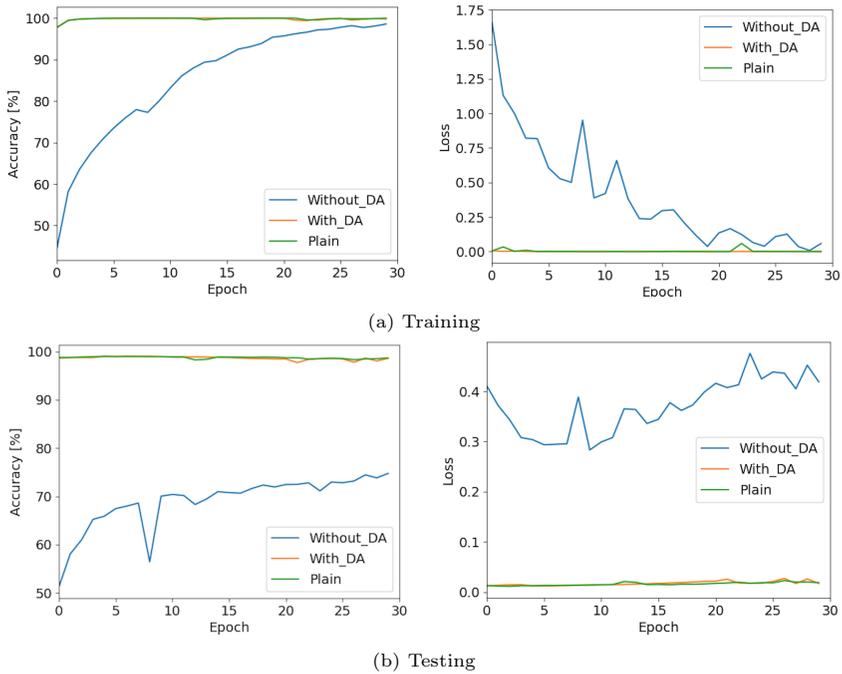


Figure 6: Learning curves during training and testing

to 30. For training, from the figure, a model fine-tuned with only encrypted images (without\_DA) needed a larger number of epochs to achieve the same values as those of the plain model (Plain). In contrast, when using the domain adaptation, the model fine-tuned with encrypted images (Proposed) achieved almost the same performances in each epoch as the model trained with plain images. This means that even if encrypted images are used, the domain adaptation allows us not only to train privacy-preserving models with almost the same accuracy as plain models but to also avoid increasing the model training time.

For testing, the model without the domain adaptation was confirmed to have degraded accuracy and loss performances compared with the plain model. In contrast, the model with domain adaptation had almost the same performances as the plain model.

Accordingly, the proposed method allows us not only to train privacy-preserving models without performance degradation but to also avoid an increase in training time even when using encrypted images.

#### 4.4 Comparison with Conventional Methods

In Tables 2 and 3, the proposed method is compared with the state-of-the-art methods for privacy preserving image classification with encrypted images such as LE [20], PE [19], and ELE [13]. From the tables, the proposed method not only has the lowest performance degradation on all datasets but also outperforms all existing methods in terms of the image classification accuracy.

Table 2: Comparison with existing methods (CIFAR-10 and CIFAR-100)

Encryption	Model	Accuracy (%) $\uparrow$	
		CIFAR-10	CIFAR-100
LE [20]	Shakedrop + AdaptNet	94.49	75.48
	ViT-B_16	91.37	67.95
ELE [13]	Shakedrop + AdaptNet	83.06	62.97
	ViT-B_16	40.64	17.15
PE [19]	ResNet-18	92.03	-
	ViT-B_16	87.57	67.97
EtC [1]	ViT-B_16	87.89	-
	ConvMixer-256/8	92.72	-
Color-NeuraCrypt [17]	ViT-B_16	96.20	-
Block scrambling + Pixel position shuffling [16]	ViT-B_16	96.64	84.42
Block scrambling + Pixel shuffling + Negative/Positive transformation [18]	ConvMixer-512/16 + Adaptive matrix	92.65	-
Block scrambling + Pixel shuffling + Domain adaptation (Proposed)	ViT-B_16	<b>98.98</b>	<b>92.58</b>
Plain	ShakeDrop	96.70	83.59
	ResNet-18	95.53	-
	ViT-B_16	99.00	92.60
	ConvMixer-256/8	96.07	-

Table 3: Comparison with existing methods (Imagenette)

Encryption	Model	Accuracy (%) $\uparrow$
Block scrambling + Pixel shuffling + Domain adaptation (Proposed)	ViT-B_16	<b>99.2</b>
EtC [1]	ViT-B_16	90.62
	ConvMixer-384/8	90.11
Plain	ViT-B_16	99.2
	ConvMixer-384/8	96.10

## 5 Conclusion

In this paper, we proposed a domain adaptation method to reduce the influence of image encryption for privacy-preserving ViT. The method is applied to a model pre-trained in the plain domain prior to the fine-tuning of the pre-trained model. The domain adaptation is carried out with two random matrices generated with secret keys, and test images are also encrypted by using the keys. In experiments, the method was demonstrated not only to reduce the performance degradation of models but to also achieve the highest classification accuracy among conventional methods on the CIFAR-10, CIFAR-100, and Imagenette datasets.

## References

- [1] M. AprilPyone and H. Kiya, “Privacy-Preserving Image Classification Using an Isotropic Network”, *IEEE MultiMedia*, 29(2), 2022, 23–33, DOI: [10.1109/MMUL.2022.3168441](https://doi.org/10.1109/MMUL.2022.3168441).
- [2] M. AprilPyone and H. Kiya, “Block-Wise Image Transformation With Secret Key for Adversarially Robust Defense”, *IEEE Transactions on Information Forensics and Security*, 16, 2021, 2709–23, DOI: [10.1109/TIFS.2021.3062977](https://doi.org/10.1109/TIFS.2021.3062977).
- [3] A. H. Chang and B. M. Case, “Attacks on Image Encryption Schemes for Privacy-Preserving Deep Neural Networks”, 2020, arXiv: [2004.13263](https://arxiv.org/abs/2004.13263) [[cs.CR](https://arxiv.org/abs/2004.13263)].
- [4] T. Chuman, W. Sirichotedumrong, and H. Kiya, “Encryption-Then-Compression Systems Using Grayscale-Based Image Encryption for JPEG Images”, *IEEE Transactions on Information Forensics and Security*, 14(6), 2019, 1515–25, DOI: [10.1109/TIFS.2018.2881677](https://doi.org/10.1109/TIFS.2018.2881677).
- [5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7*, OpenReview.net, 2021.
- [6] J. Howard, “imagenette”, <https://github.com/fastai/imagenette/>.
- [7] H. Ito, Y. Kinoshita, M. Aprilpyone, and H. Kiya, “Image to Perturbation: An Image Transformation Network for Generating Visually Protected Images for Privacy-Preserving Deep Neural Networks”, *IEEE Access*, 9, 2021, 64629–38, DOI: [10.1109/ACCESS.2021.3074968](https://doi.org/10.1109/ACCESS.2021.3074968).
- [8] H. Kiya, M. AprilPyone, Y. Kinoshita, S. Imaizumi, and S. Shiota, “An Overview of Compressible and Learnable Image Transformation with Secret Key and its Applications”, *APSIPA Transactions on Signal and Information Processing*, 11(1, e11), 2022, DOI: [10.1561/116.00000048](https://doi.org/10.1561/116.00000048).
- [9] H. Kiya, R. Iijima, M. AprilPyone, and Y. Kinoshita, “Image and Model Transformation with Secret Key for Vision Transformer”, *IE-ICE Transactions on Information and Systems*, E106.D(1), 2023, 2–11, DOI: [10.1587/transinf.2022MUI0001](https://doi.org/10.1587/transinf.2022MUI0001).
- [10] H. Kiya, T. Nagamori, S. Imaizumi, and S. Shiota, “Privacy-Preserving Semantic Segmentation Using Vision Transformer”, *Journal of Imaging*, 8(9), 2022, ISSN: 2313-433X, DOI: [10.3390/jimaging8090233](https://doi.org/10.3390/jimaging8090233), <https://www.mdpi.com/2313-433X/8/9/233>.
- [11] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated Learning: Strategies for Improving Communication Efficiency”, *arXiv*, 2016, <https://arxiv.org/abs/1610.05492>.

- [12] A. Krizhevsky, “Learning multiple layers of features from tiny images”, in, University of Toronto, Tech. Rep., 2009.
- [13] K. Madono, M. Tanaka, O. Masaki, and O. Tetsuji, “Block-wise Scrambled Image Recognition Using Adaptation Network”, in *Workshop on Artificial Intelligence of Things (AAAI WS)*, 2020.
- [14] M. Maung, A. Pyone, and H. Kiya, “Encryption Inspired Adversarial Defense For Visual Classification”, in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, 1681–5, DOI: [10.1109/ICIP40778.2020.9190904](https://doi.org/10.1109/ICIP40778.2020.9190904).
- [15] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, “Privacy-Preserving Deep Learning via Additively Homomorphic Encryption”, *IEEE Transactions on Information Forensics and Security*, 13(5), 2018, 1333–45, DOI: [10.1109/TIFS.2017.2787987](https://doi.org/10.1109/TIFS.2017.2787987).
- [16] Z. Qi, M. AprilPyone, Y. Kinoshita, and H. Kiya, “Privacy-Preserving Image Classification Using Vision Transformer”, *EURASIP European Signal Processing Conference, Belgrade, Serbia, August 31, 2022*, 543–7.
- [17] Z. Qi, M. AprilPyone, and H. Kiya, “Color-NeuraCrypt: Privacy-Preserving Color-Image Classification Using Extended Random Neural Networks”, 2023, arXiv: [2301.04875](https://arxiv.org/abs/2301.04875).
- [18] Z. Qi, A. MaungMaung, and H. Kiya, “Privacy-Preserving Image Classification Using ConvMixer with Adaptive Permutation Matrix and Block-Wise Scrambled Image Encryption”, *Journal of Imaging*, 9(4), 2023, ISSN: 2313-433X, <https://www.mdpi.com/2313-433X/9/4/85>.
- [19] W. Sirichotedumrong, Y. Kinoshita, and H. Kiya, “Pixel-Based Image Encryption Without Key Management for Privacy-Preserving Deep Neural Networks”, *IEEE Access*, 7, 2019, 177844–55, DOI: [10.1109/ACCESS.2019.2959017](https://doi.org/10.1109/ACCESS.2019.2959017).
- [20] M. Tanaka, “Learnable Image Encryption”, in *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 2018, 1–2, DOI: [10.1109/ICCE-China.2018.8448772](https://doi.org/10.1109/ICCE-China.2018.8448772).
- [21] A. Trockman and J. Z. Kolter, “Patches Are All You Need?”, *arXiv*, abs/2201.09792, 2022, DOI: [10.48550/ARXIV.2201.09792](https://doi.org/10.48550/ARXIV.2201.09792).
- [22] Y. Wang, J. Lin, and Z. Wang, “An Efficient Convolution Core Architecture for Privacy-Preserving Deep Learning”, in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, 1–5, DOI: [10.1109/ISCAS.2018.8350963](https://doi.org/10.1109/ISCAS.2018.8350963).
- [23] K. Wei, J. Li, C. Ma, M. Ding, W. Chen, J. Wu, M. Tao, and H. V. Poor, “Personalized Federated Learning With Differential Privacy and Convergence Guarantee”, *IEEE Transactions on Information Forensics and Security*, 18, 2023, 4488–503, DOI: [10.1109/TIFS.2023.3293417](https://doi.org/10.1109/TIFS.2023.3293417).

- [24] A. Yala, H. Esfahanizadeh, R. G. L. D. Oliveira, K. R. Duffy, M. Ghobadi, T. S. Jaakkola, V. Vaikuntanathan, R. Barzilay, and M. Medard, “NeuraCrypt: Hiding Private Health Data via Random Neural Networks for Public Training”, 2021, arXiv: [2106.02484](https://arxiv.org/abs/2106.02484).