information

MDPI

Article

Privacy-Preserving ConvMixer Without Any Accuracy Degradation Using Compressible Encrypted Images

Haiwei Lin [1], **Shoko Imaizumi** [2,*] and **Hitoshi Kiya** [3,*]

[1] Graduate School of Science and Engineering, Chiba University, Chiba 263-8522, Japan; 23wm3202@student.gs.chiba-u.jp
[2] Graduate School of Informatics, Chiba University, Chiba 263-8522, Japan
[3] Faculty of System Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan
[*] Correspondence: imaizumi@chiba-u.jp (S.I.); kiya@tmu.ac.jp (H.K.)

Abstract: We propose an enhanced privacy-preserving method for image classification using ConvMixer, which is an extremely simple model that is similar in spirit to the Vision Transformer (ViT). Most privacy-preserving methods using encrypted images cause the performance of models to degrade due to the influence of encryption, but a state-of-the-art method was demonstrated to have the same classification accuracy as that of models without any encryption under the use of ViT. However, the method, in which a common secret key is assigned to each patch, is not robust enough against ciphertext-only attacks (COAs) including jigsaw puzzle solver attacks if compressible encrypted images are used. In addition, ConvMixer is less robust than ViT because there is no position embedding. To overcome this issue, we propose a novel block-wise encryption method that allows us to assign an independent key to each patch to enhance robustness against attacks. In experiments, the effectiveness of the method is verified in terms of image classification accuracy and robustness, and it is compared with conventional privacy-preserving methods using image encryption.

Keywords: privacy preserving; image classification; patch embedding; image encryption; access control

check for updates

Citation: Lin, H.; Imaizumi, S.; Kiya, H. Privacy-Preserving ConvMixer Without Any Accuracy Degradation Using Compressible Encrypted Images. *Information* **2024**, *15*, 723. https://doi.org/10.3390/info15110723

Academic Editors: Nikolaos Mitianoudis and Ilias Theodorakopoulos

Received: 7 October 2024
Revised: 6 November 2024
Accepted: 7 November 2024
Published: 11 November 2024

1. Introduction

Deep neural networks (DNNs) have been deployed in many applications including security-critical ones such as biometric authentication and medical image 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 trusted in general, privacy-preserving deep learning has become an urgent problem [1–9].

One of the privacy-preserving solutions for DNNs is to use encrypted images to protect visual information in images for testing models. In this approach, which has been inspired by various compressible encryption methods [1,10,11], images are transformed by using a secret key, and images transformed by using a perceptual encryption method are used as testing data. However, most conventional methods [12–14] have a problem, that is, the performance of encrypted models degrades compared with models without encryption. Conventional cryptographic methods such as homomorphic encryption [15–19] are one of the other privacy-preserving approaches, but the computational cost of implementation is high, and it is not easy to apply these methods to state-of-the art DNNs directly. In contrast, privacy-preserving federated learning [16,20,21] 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 a use of encrypted images that does not degrade the performance of models. It was also pointed out that the use of an isotropic network such as the Vision Transformer (ViT) [22] can avoid the performance degradation of models under some requirements [23] even when encrypted images are applied to a model. Nevertheless, the encrypted images derived from [23] are vulnerable to an extended jigsaw puzzle solver (EJPS) attack [24]. The EJPS attack can effectively restore visual information from the encrypted images, especially when compressible image encryption methods are applied. This vulnerability is more noticeable when the model is replaced with ConvMixer [25]. ConvMixer is another isotropic network, demonstrating comparable performance with fewer parameters than ViT. However, in contrast to ViT, ConvMixer does not have position embedding to handle patches in a random order. Thus, block scrambling cannot be applied to the encrypted images in this case. This increases vulnerability to the EJPS attack, since block scrambling plays a critical role in obfuscating visual information.

In this paper, to overcome this issue, an enhanced privacy-preserving method is proposed for image classification using ConvMixer by extending the key assignment used for image encryption. In experiments, the proposed method is verified not only to enhance robustness against ciphertext-only attacks (COAs) including the EJPS attack but to also maintain the same classification accuracy as that of models without encryption on the CIFAR-10 dataset [26].

We make the following contributions in this paper.

(a) We propose a novel image encryption method that allows us to use independent keys for each patch to enhance resistance against attacks.

(b) We verify that the proposed sub-block-wise encryption using independent keys is more robust that conventional ones even against the state-of-the-art attack, EJPS, while maintaining the same classification accuracy as that of using plain images.

The rest of this paper is structured as follows. Section 2 presents related works on image encryption for deep learning and ConvMixer. Regarding the proposed method, Section 3 gives an overview and includes image encryption, model encryption, and security analysis. Experiments for verifying the effectiveness of the method, including classification accuracy and robustness against attacks, are presented in Section 4, and Section 5 concludes this paper.

2. Related Work

Image encryption methods for deep learning and ConvMixer are summarized here.

2.1. Image Encryption for Deep Learning

Image transformation methods using a secret key, often referred to as perceptual image encryption or image cryptography, have been studied so far for various applications. Image encryption enables us not only to protect the visual information of plain images but also to embed unique features controlled with the key into images. One of the origins of image transformation with a key is in block-wise image encryption, that is, compressible encryption for encryption-then-compression (EtC) systems [1,10]. In addition, encrypted data have been demonstrated to be effective in privacy-preserving learning [10,12–14,27–29], adversarial defense [30], and access control [31].

Tanaka first introduced a block-wise learnable image encryption method (LE) with an adaptation layer [12], 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 an adaptation layer [32]. However, these two encryption methods are not robust enough against ciphertext-only attacks (COAs), as reported in [33,34]. To enhance the security of encryption, LE was improved to an extended learnable image encryption method (ELE) [13] by adding a block scrambling (permutation) step and a pixel encryption operation with multiple keys. However, ELE has an inferior accuracy compared with using plain images, even when an additional adaptation network is applied to reduce the influence of the encryption.

Recently, block-wise encryption was also pointed out to have a high similarity with isotropic networks such as ViT [22] and ConvMixer [25], and this similarity enables us to reduce performance degradation [20], but these methods still have the same performance degradation problem as conventional block-wise encryption methods. In contrast, it was pointed out that the use of an isotropic network such as ViT and ConvMixer can avoid the performance degradation of models under some requirements [23] even when encrypted images are applied to a model. However, the EJPS attack [24] was demonstrated to restore visual information in images from encrypted ones when compressible encrypted images are applied to a model.

To overcome this issue, we propose an enhanced privacy-preserving method using compressible encrypted images for image classification with ConvMixer.

2.2. ConvMixer

ConvMixer is a convolutional neural network (CNN) with patch embedding inspired by ViT. Despite its simplicity, ConvMixer outperforms standard vision models, including ViT, ResNet, and some of their variants for similar parameter counts and dataset scales [35].

As shown in Figure 1, a standard ConvMixer consists of a patch embedding followed by L ConvMixer layers, where each ConvMixer layer is composed of a depthwise convolution block and a pointwise convolution block.

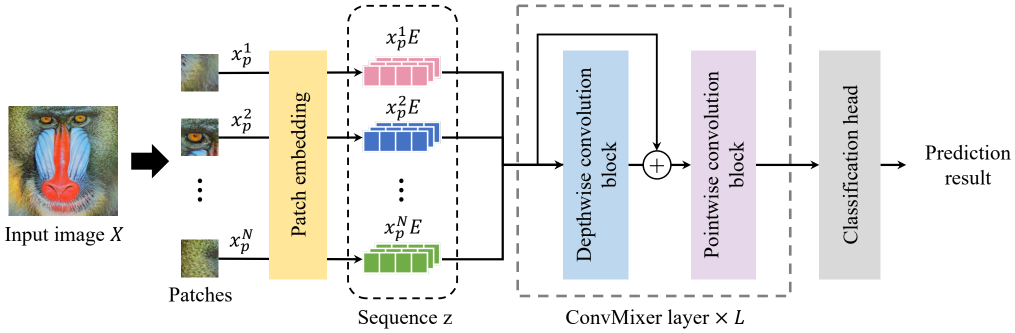


Figure 1. Standard inference pipeline of ConvMixer for image classification.

Similar to ViT, in a standard inference pipeline of ConvMixer, an input image $X \in \mathbb{R}^{H \times W \times C}$ is first divided into N non-overlapping patches with a size of $P \times P$. Here, H, W, and C are the height, width, and channel number of the input image, respectively. Thus, the total number of patches N can be given by $N = (HW/P)^2$. To convert the patches to a valid input to the first ConvMixer layer, each patch is processed through patch embedding expressed as

$$z = [x_p^1E, x_p^2E, \dots, x_p^iE, \dots, x_p^NE]. \tag{1}$$

In the above equation, each patch is flattened into a vector denoted by $x_p^i \in \mathbb{R}^{P^2C}$, where $i \in \{1, 2, \dots, N\}$. Each x_p^i is subjected to a linear transformation with a matrix $E \in \mathbb{R}^{P^2C \times D}$, resulting in a D-dimensional vector x_p^iE. Here, E is a learnable weight. Finally, all the D-dimensional vectors are integrated into a sequence denoted by z, serving as the input to the first ConvMixer layer. z passes through L ConvMixer layers to the classification head, from which a prediction result is output.

As shown in Equation (1), ConvMixer does not have position embedding due to its CNN backbone, but ViT does. The lack of position embedding makes it difficult for ConvMixer to handle EtC images with block scrambling (BS), so EtC images for ConvMixer are less robust against attacks than those for ViT.

3. Proposed Method

3.1. Overview of Proposed Method

Figure 2 shows the framework of privacy-preserving image classification using encrypted images, where an untrusted cloud provider has an encrypted ConvMixer as a model, which is provided by a model developer. The model developer encrypts a model trained with plain images by using a key chain.

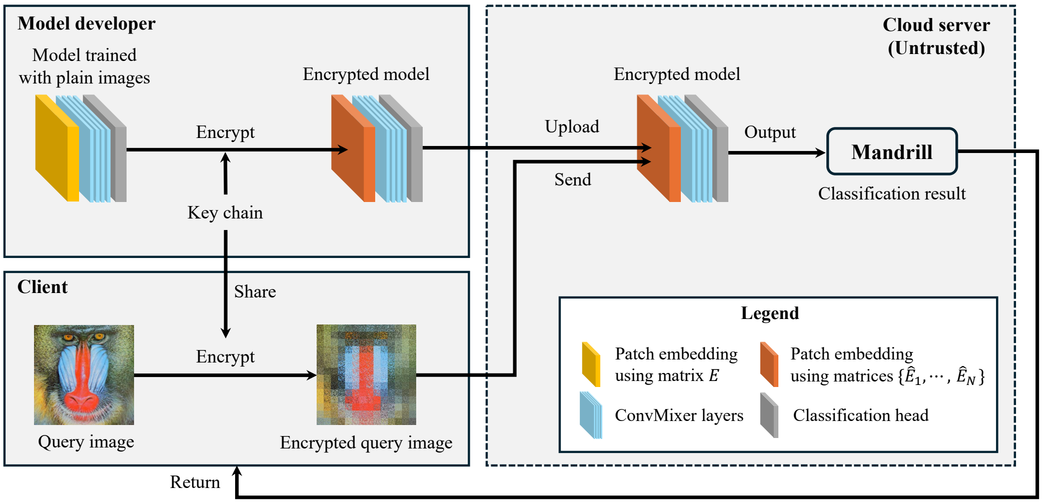


Figure 2. Framework of privacy-preserving image classification using encrypted images.

Next, to prevent unauthorized clients or attackers from maliciously using the model, an authorized client encrypts query images with the same key chain as that of the model encryption and sends the encrypted ones to the server. This server does not have both the keys and the visual information of the query images. If the encrypted images are compressible, called EtC images, the clients can send compressed data to the server.

Using the above framework, we make the following contributions in this paper (see Figure 3). We propose a novel image encryption method that allows us to use independent keys for each patch to enhance resistance against attacks. In addition, the proposed sub-block-wise encryption using independent keys is demonstrated to be robust even against the state-of-the-art attack, EJPS, while maintaining the same classification accuracy as that of using plain images.

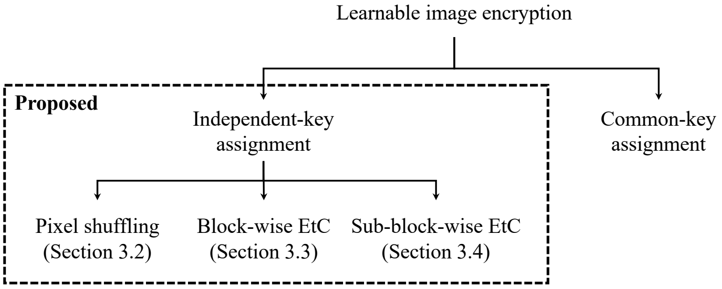

Learnable image encryption
Proposed
Independent-key assignment
Common-key assignment
Pixel shuffling (Section 3.2)
Block-wise EtC (Section 3.3)
Sub-block-wise EtC (Section 3.4)

Figure 3. Overview of proposed method.

3.2. Image Encryption with Block-Wise Pixel Shuffling

In this section, we elaborate on the block-wise image encryption mentioned in Figure 3. This image encryption obfuscates an image block by block. However, in previous works, the encryption of each block was guided by a single common key. In contrast, we develop a novel key assignment method that uses an independent key for each block. To help understand our method, we present a fundamental procedure based on block-wise pixel shuffling, in which encrypted images are not compressible. The procedure is given as follows.

Step 1: Divide a query image $X \in \mathbb{R}^{H \times W \times C}$ into N non-overlapping blocks with a size of $P \times P$ as follows:

$$B = \{B_1, \ldots, B_i, \ldots, B_N\}, \tag{2}$$

where each block $B_i \in \mathbb{R}^{P \times P \times C}$, and $i \in \{1, 2, \ldots, N\}$. Note that N and P are identical to the number of patches and the patch size in patch embedding, respectively.

Step 2: Prepare a key chain K, where K is expressed as $K = \{k_1, \ldots, k_i, \ldots, k_N\}$, where $k_i \in K$ denotes a random permutation matrix with a size of $P^2C \times P^2C$, defined as

$$k_i = \begin{bmatrix} k_i(1,1) & k_i(1,2) & \cdots & k_i(1,P^2C) \\ k_i(2,1) & k_i(2,2) & \cdots & k_i(2,P^2C) \\ \vdots & \vdots & \ddots & \vdots \\ k_i(P^2C,1) & k_i(P^2C,2) & \cdots & k_i(P^2C,P^2C) \end{bmatrix}. \tag{3}$$

Step 3: Flatten each block $B_i \in B$ into a vector x_p^i given by

$$x_p^i = [x_p^i(1), \ldots, x_p^i(j), \ldots, x_p^i(P^2C)]. \tag{4}$$

Here, $x_p^i(j)$ represents a pixel in B_i.

Step 4: Using permutation matrix k_i, shuffle pixel positions within vector x_p^i by

$$\hat{x}_p^i = x_p^i k_i, \tag{5}$$

where $\hat{x}_p^i$ denotes the encrypted form of x_p^i.

Step 5: Reshape each $\hat{x}_p^i$ to a block with a size of $P \times P$ and integrate the blocks into an encrypted query image $\hat{X}$.

In previous methods, a single common key that met $k_1 = k_2 = \ldots = k_N$ was assigned to each block. In contrast, in the proposed method, independent keys that meet $k_1 \neq k_2 \neq \ldots \neq k_N$ are applied to generate encrypted images. As described later, the novel key assignment can enhance the security of encrypted images.

3.3. Image Encryption with Block-Wise EtC

We introduced a novel image encryption method with independent keys for ConvMixer in Section 3.2, but the encrypted images are not compressible. Accordingly, we propose generating block-wise EtC images with independent keys. In general, block-wise EtC images are generated by using four block-wise transformations, namely, rotation and flip, negative–positive inversion, channel shuffling, and block scrambling, but ConvMixer does not allow us to apply block scrambling for image encryption because sequence z in Equation (1) does not have position embedding. Due to this limitation, the EtC images used for ConvMixer are not robust enough against attacks.

To address the issue, two strategies are applied to EtC images for ConvMixer in this paper. The first one is the key assignment method presented in Section 3.2, and the second one is a sub-block-wise operation, which will be discussed in Section 3.4. Prior to sub-block-wise EtC, block-wise EtC using independent keys is explained below (see Figure 4).

Step 1: Divide an image $X \in \mathbb{R}^{C \times H \times W}$ into N non-overlapping blocks with a size of $P \times P$. These blocks are represented as $B = \{B_1, \ldots, B_i, \ldots, B_N\}$.

Step 2: Prepare a key chain consisting of N keys given by $K = \{k_1, \ldots, k_i, \ldots, k_N\}$. In this case, each key $k_i \in K$ is defined as $k_i = \{k_i^1, k_i^2, k_i^3\}$, where $k_i^1 \in \{1, 2, \ldots, 8\}$, $k_i^2 \in \{1, 2\}$, and $k_i^3 \in \{1, 2, \ldots, 6\}$.

Step 3: Apply the following transformations to block B_i:

(a) Randomly rotate and flip block B_i with the pattern provided in Figure 5, where the pattern is decided on $k_i^1 \in k_i$.

(b) Randomly invert the pixel values of each block obtained in Step 3 (a) with the pattern provided in Figure 6, where the pattern is decided on $k_i^2 \in k_i$.

(c) Randomly permute the order of color channels in the block obtained in Step 3 (b) as in Figure 7, where the order is decided on $k_i^3 \in k_i$.

Step 4: Integrate all the blocks into an image to obtain an encrypted image $\hat{X}$.

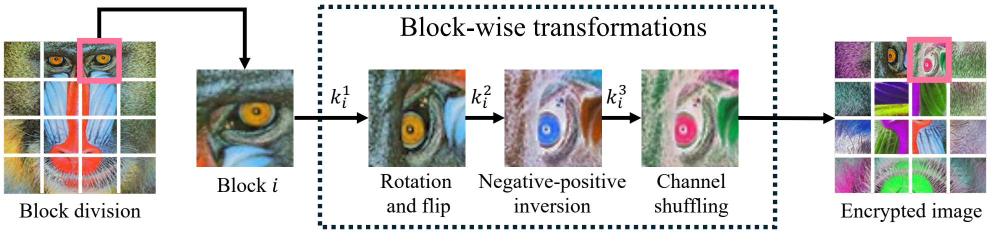


Figure 4. Encryption pipeline of block-wise EtC.

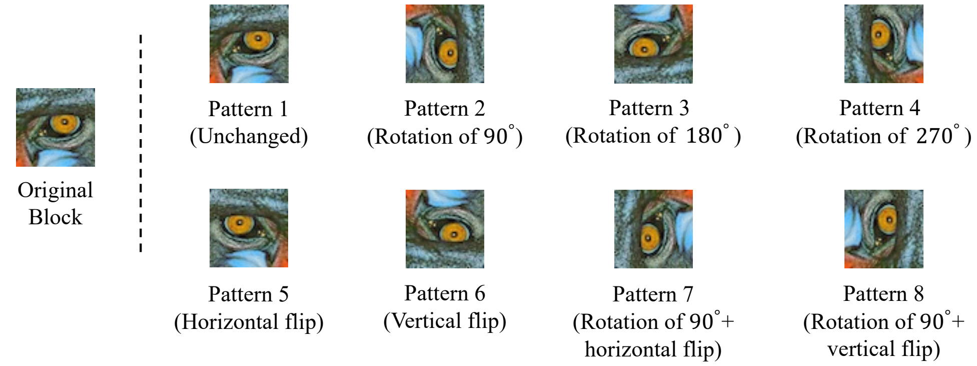

Original Block
Pattern 1 (Unchanged)
Pattern 2 (Rotation of 90°)
Pattern 3 (Rotation of 180°)
Pattern 4 (Rotation of 270°)
Pattern 5 (Horizontal flip)
Pattern 6 (Vertical flip)
Pattern 7 (Rotation of 90°+ horizontal flip)
Pattern 8 (Rotation of 90°+ vertical flip)

Figure 5. Possible patterns for rotation and flip.

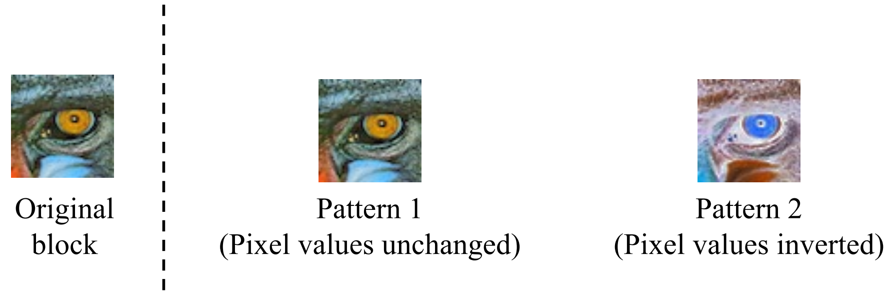

Original block
Pattern 1 (Pixel values unchanged)
Pattern 2 (Pixel values inverted)

Figure 6. Possible patterns for negative–positive inversion.

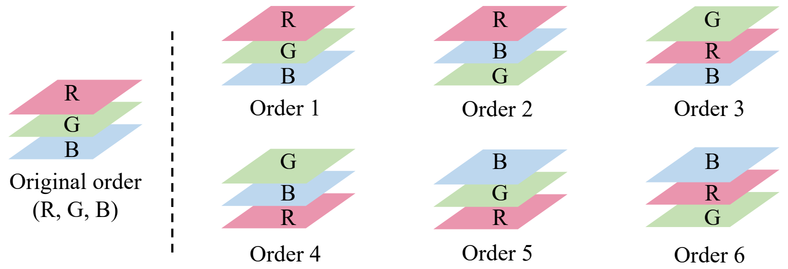


Figure 7. Random order permutation of three channels.

3.4. Image Encryption with Sub-Block-Wise EtC

Block-wise EtC images for ConvMixer are generated by using three of the block-wise transformations: rotation and flip, negative–positive inversion, and channel shuffling. To add sub-block scrambling for image encryption, sub-block-wise EtC is proposed here. Sub-block-wise EtC is an extension of block-wise EtC and conducts transformations on the sub blocks divided from a block.

Figure 8 shows the procedure for generating sub-block-wise EtC images. In this figure, a given image is encrypted with the following steps:

Step 1: Divide a given image into N blocks with $P \times P$ pixels. These blocks are denoted by $B = \{B_1, \ldots, B_i, \ldots, B_N\}$.

Step 2: Divide each block B_i into N_s sub blocks with $S \times S$ pixels as

$$B_i = \{b_i^1, \ldots, b_i^j, \ldots, b_i^{N_s}\}, \tag{6}$$

where $N_s = (P/S)^2$.

Step 3: Prepare a key chain $K = \{k_1, \ldots, k_i, \ldots, k_N\}$, where $k_i \in K$ is expressed as

$$k_i = \{k_i^1, k_i^2, k_i^3, k_i^4\}. \tag{7}$$

Here, k_i^1, k_i^2, k_i^3, and k_i^4 are given by

$$\begin{aligned} k_i^1 &= \{\alpha_i^1, \ldots, \alpha_i^j, \ldots, \alpha_i^{N_s}\}, \\ k_i^2 &= \{\beta_i^1, \ldots, \beta_i^j, \ldots, \beta_i^{N_s}\}, \\ k_i^3 &= \{\gamma_i^1, \ldots, \gamma_i^j, \ldots, \gamma_i^{N_s}\}, \\ k_i^4 &= \{\delta_i^1, \ldots, \delta_i^j, \ldots, \delta_i^{N_s}\}, \end{aligned} \tag{8}$$

where $\alpha_i^j \in \{1, 2, \ldots, 8\}$, $\beta_i^j \in \{1, 2\}$, $\gamma_i^j \in \{1, 2, \ldots, 6\}$, and $\delta_i^j \in \{1, 2, \delta_i^j, \ldots, \delta_i^l, \ldots, N_s\}$. Note that

$$\delta_i^j \neq \delta_i^l, \quad if \quad j \neq l. \tag{9}$$

Step 4: Using k_i^1, k_i^2, k_i^3, and k_i^4, encrypt sub block b_i^j in accordance with the following procedure:

(a) Randomly rotate and flip sub block b_i^j with the pattern provided in Figure 5, where the pattern is decided on $\alpha_i^j \in k_i^1$.

(b) Randomly invert the pixel values of the sub block obtained in Step 4 (a) with the pattern provided in Figure 6, where the pattern is decided on $\beta_i^j \in k_i^2$.

(c) Randomly permute the order of color channels in the sub block obtained in Step 4 (b) as in Figure 7, where the order is decided on $\gamma_i^j \in k_i^3$.

Step 5: Randomly permute the positions of sub blocks in the block obtained in Step 4 in accordance with $\delta_i^j \in k_i^4$. Figure 9 shows an example of the sub-block scrambling. Finally, an encrypted image can be obtained by integrating all the blocks.

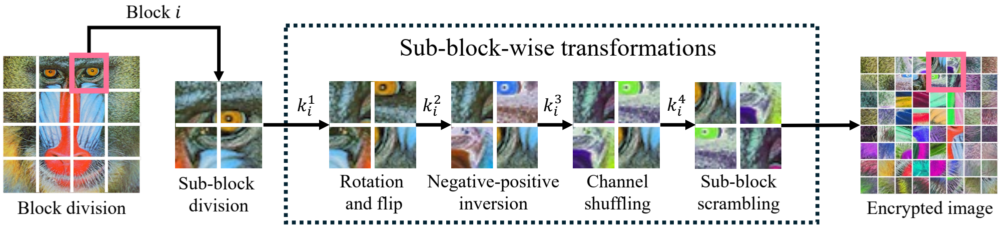


Figure 8. Generation of sub-block-wise EtC images.

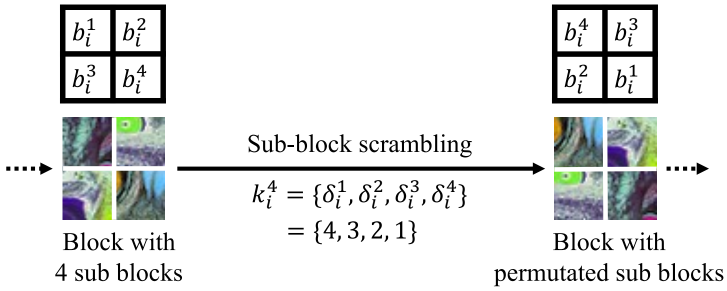


Figure 9. Example of sub-block scrambling ($N_s = 4$).

3.5. Model Encryption

As shown in Figure 2, a model trained with plain images is encrypted by using the same key chain as that used for image encryption. The model encryption is applied to Equation (1). Other parameters in the model are the same as those of plain model. For example, for block-wise pixel shuffling, E in Equation (1) is transformed with k_i in Equation (3) as

$$\hat{E}_i = k_i^\top E, \tag{10}$$

where k_i is a random permutation matrix, which is a square binary matrix that has exactly one entry of 1 in each row and each column, with all other entries being 0. Every permutation matrix is orthogonal, with its inverse equal to its transpose. Accordingly, when encrypted query images are input to the encrypted model, a sequence of embedded patches is given by

$$\begin{aligned}\hat{z} &= [\hat{x}_p^1\hat{E}_1, \hat{x}_p^2\hat{E}_2, \ldots, \hat{x}_p^i\hat{E}_i, \ldots, \hat{x}_p^N\hat{E}_N] \\ &= [x_p^1 k_1 k_1^\top E, x_p^2 k_2 k_2^\top E, \ldots, x_p^i k_i k_i^\top E, \ldots, x_p^N k_N k_N^\top E] \\ &= [x_p^1 E, x_p^2 E, \ldots, x_p^i E, \ldots, x_p^N E] \\ &= z\end{aligned}. \tag{11}$$

From the above equation, we can confirm that the classification accuracy when using encrypted models is the same as that of using original models without any encryption. k_i used for block-wise EtC and sub-block-wise EtC can also be given a random permutation matrix, so the proposed method does not cause any accuracy degradation even when encrypted images are used.

3.6. Security Enhancement

Here, we discuss the security strength of our method in terms of key space. Key space refers to the total number of possible keys that can be used in an encryption algorithm. The size of the key space directly affects the robustness against brute force attacks that attempt to try all possible keys so that the correct one can be found. Generally, a key space of 2^{256} or larger is recommended for strong security.

In the block-wise EtC images, a query image is divided into N blocks, each of which is subjected to three block-wise transformations. Using our method, each block is encrypted with a key independent of other blocks. Therefore, when independent keys are assigned to N blocks, the key space is

$$O_b^{ind} = (8 \times 2 \times 6)^N, \tag{12}$$

where 8, 2, and 6 represent key spaces for rotation and flip, negative–positive inversion, and channel shuffling, respectively. Similarly, in the sub-block-wise EtC, the key space can be calculated as

$$O_{sb}^{ind} = 8^{N_s} \times 2^{N_s} \times 6^{N_s} \times N_s!. \tag{13}$$

In this case, since each transformation is performed in a sub-block-wise manner, the key spaces for rotation and flip, negative–positive inversion, and channel shuffling are rewritten as 8^{N_s}, 2^{N_s}, and 6^{N_s}, respectively. $N_s!$ represents the key space for sub-block scrambling.

We assume that 224 × 224 query images are compressed with the JPEG standard, where the block size and sub-block size are set to 16 × 16 and 8 × 8, respectively. For this setting, Table 1 shows a key space comparison among four types of EtC methods. From the table, it is evident that the key space is significantly smaller than the recommended size when adopting the common key assignment, regardless of whether block-wise EtC or sub-block-wise EtC is used. This makes the encrypted images highly vulnerable to brute force attacks. In contrast, when our independent key assignment is used, the key space is enlarged exponentially. In other words, the robustness against brute force attacks is greatly enhanced. To comprehensively evaluate the security, we further assess the robustness of encrypted images using the EJPS attack in the following experiments.

The EJPS attack was demonstrated to restore visual information from sub-block-wise images encrypted with a common key under the use of ConvMixer, so sub-block-wise images encrypted with independent keys have to be robust against the attack. The EJPS attack is the strongest attack against EtC images, and it mainly involves two steps: sub-block restoration and jigsaw-puzzle solution. The first step restores each block by analyzing the edge pixel correlation among the sub blocks in a block, and the restored blocks are then reassembled into an image through the second step. Therefore, using an independent key for each block is expected to significantly increase the complexity of the first step, improving robustness against the EJPS attack. In the next section, we will verify the effectiveness of our proposed method by simulating a scenario involving the EJPS attack.

Table 1. Key space of EtC images. Note that N denotes the number of blocks in the query image, and N_s denotes the number of sub blocks in each block.

Method	Key Assignment	Key Space ($N = 196$, $N_s = 4$)
Block-wise EtC	Common	$8 \times 2 \times 6 \ll 2^{256}$
Block-wise EtC (Proposed)	Independent	$(8 \times 2 \times 6)^{196} \gg 2^{256}$
Sub-block-wise EtC	Common	$8^4 \times 2^4 \times 6^4 \times 4! \ll 2^{256}$
Sub-block-wise EtC (Proposed)	Independent	$(8^4 \times 2^4 \times 6^4 \times 4!)^{196} \gg 2^{256}$

4. Experiments

In experiments, the effectiveness of our method is verified in terms of the classification accuracy of the method and the security of sub-block-wise EtC images.

4.1. Experimental Setup

Our experiment was conducted with the CIFAR-10 dataset [26]. To be equal to the standard input size of ConvMixer, each image was resized to 224 × 224 pixels by bicubic interpolation. We compared the proposed method with the previous method using a common key. An example of encrypted images is shown in Figure 10, where Com-16 and Ind-16 are encryption methods using block-wise EtC with a common key and independent keys, and Com-16/8 and Ind-16/8 are those using sub-block-wise EtC with a common key and independent keys, respectively. For instance, in Com-16/8, the block and sub-block sizes are 16 and 8, respectively.

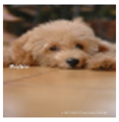
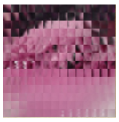
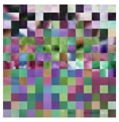
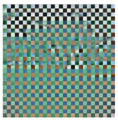
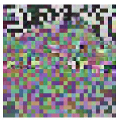
	Plain	Com-16	Ind-16	Com-16/8	Ind-16/8
Image sample					
Key assignment	–	Common	Independent	Common	Independent
Transformation	–	Block-wise	Block-wise	Sub-block-wise	Sub-block-wise
Block size	–	16×16	16×16	16×16	16×16
Sub-block size	–	–	–	8×8	8×8

Figure 10. Image details for evaluation.

A baseline (plain) model used for accuracy measurements was obtained by fine-tuning a pretrained ConvMixer model on the training set of CIFAR-10. The pretrained model was trained by using the TIMM framework [36]. The patch size P adopted for the patch embedding layer in ConvMixer was 16 × 16, which was equal to the block size adopted for the image encryption. The training and testing were conducted using an NVIDIA GeForce RTX 4080 16GB GPU (NVIDIA, Santa Clara, CA, USA).

In addition, the experiments were carried out by using a PC equipped with an Intel Core i9-12900K processor (Intel, Mountain View, CA, USA) running at 3.2 GHz and having

128 GB of main memory. The programs including the EJPS attack and its processing time measurement were implemented by Python 3.11.9, utilizing the ProcessPoolExecutor class to effectively manage and execute multiple processes in parallel.

4.2. Classification Accuracy

The proposed method was evaluated in terms of the accuracy of image classification. Table 2 shows experiment results, where plain indicates that neither the query images nor the corresponding model was encrypted. From the table, even when independent keys were assigned, the proposed method could achieve the same accuracy as that of plain and common keys. Accordingly, our method was verified to have no accuracy degradation.

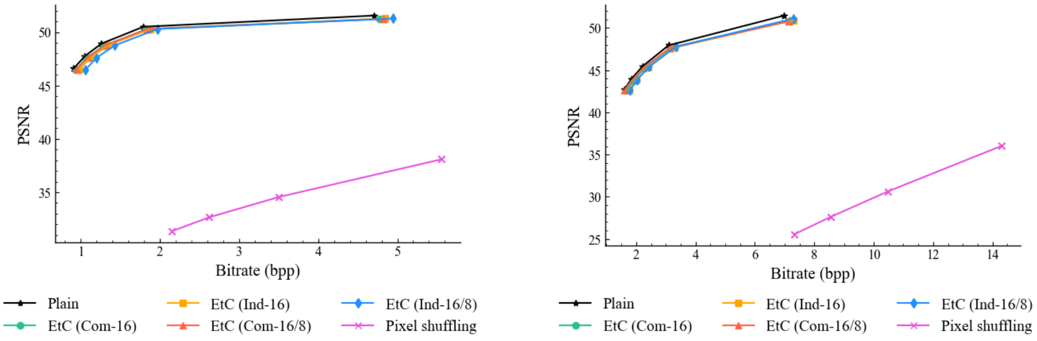
The accuracy of classification tasks depends on the dataset used in experiments, so the accuracy of plain in Table 2 will be changed if we use a dataset other than the CIFAR-10 dataset. In contrast, the accuracy for the encrypted images can remain the same as for plain as in Equation (1) even when a dataset other than the CIFAR-10 dataset is used.

Table 2. Classification accuracy.

Query Images	Classification Accuracy [%]
Plain	96.86
Com-16	96.86
Ind-16 (Proposed)	96.86
Com-16/8	96.86
Ind-16/8 (Proposed)	96.86

4.3. Compression Performance

The compression performance of various EtC images was evaluated under the use of JPEG compression, where rate–distortion (RD) curves were used to measure the compression performance. To plot RD curves, encrypted images were decrypted after decompressing the compressed ones, and peak signal-to-noise ratio (PSNR) values were then calculated. The JPEG codec provided by libjpeg [37] was used for JPEG compression, where chroma subsampling was set to 4:4:4. From Figure 11a, we can see that the EtC images were compressible unlike pixel shuffling, as shown by its curve. To ensure that this compression performance does not depend on the dataset, we conducted another evaluation using a subset of ImageNet-1K [38]. As shown in Figure 11b, the RD curves for ImageNet-1K show a similar trend to those for CIFAR-10. Thus, the compression performance of the encrypted images created by our method is comparable to that of plain images, regardless of the dataset.



(a) CIFAR-10 (b) ImageNet-1K

Figure 11. Rate–distortion curves. Each curve was calculated by using the average value of 10 images from each dataset.

4.4. Security Analysis for Block-Wise EtC Images

We evaluated the robustness of our method against the EJPS attack. The evaluation focused on two metrics: reconstruction quality and the processing time required for reconstruction. We randomly sampled nine EtC images from images encrypted with Com-16 and Ind-16, respectively. An example of images reconstructed by the EJPS attack is given in Figure 12. It is clear that the visual information of plain images was restored from the block-wise EtC images by using the EJPS attack.

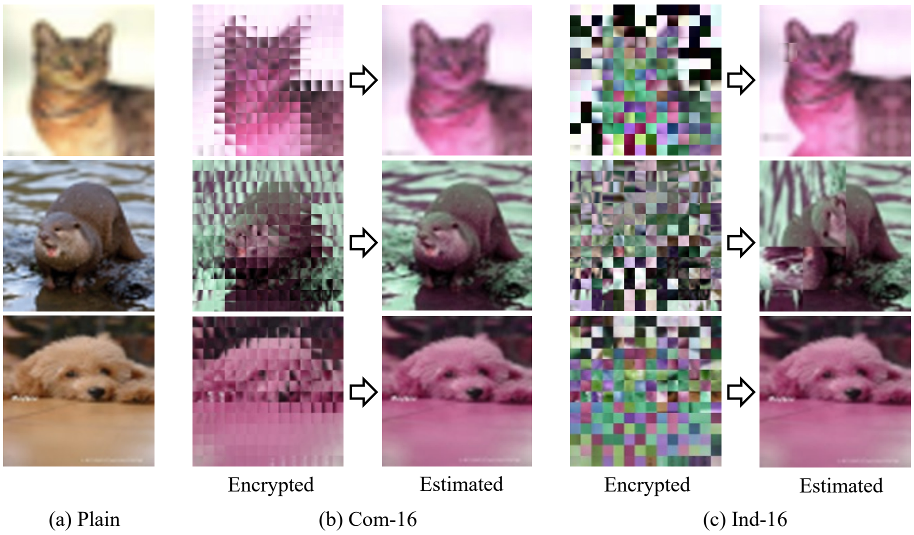


Figure 12. Results of Reconstruction by EJPS attack against block-wise EtC images.

In contrast, Table 3 shows the mean reconstruction times required for the EJPE attack. From this table, the EJPE attack required more time to reconstruct the Ind-16 images than the Com-16 ones, so the use of independent keys is effective in enhancing robustness against the EJPS attack even for block-wise EtC images.

Table 3. Computational time required for EJPS attack.

EtC Images	Average Processing Time [s]
Com-16	0.10
Ind-16	4996.94

4.5. Security Analysis for Sub-Block-Wise EtC Images

We also confirmed the robustness of sub-block-wise EtC images. We used the same nine images as those used in Section 4.4 and prepared encrypted images using Com-16/8 and Ind-16/8, respectively.

Figure 13 shows an example of restoration results for three EtC images. Most visual information of the plain images was restored from the images encrypted with Com-16/8. In contrast, the images encrypted with Ind-16/8 were not restored, so the estimated images had no identifiable information. Accordingly, the combined use of sub-block-wise EtC and independent keys can generate encrypted images that are robust against the EJPS attack. The use of smaller size images results in fewer patches, so the images are more vulnerable to the EJPS attack. Since the image size of the CIFAR-10 dataset is smaller, using these images is an evaluation under a more stringent condition. However, as shown in Figure 13, the proposed method is robust against EJPS even when using the CIFAR-10 dataset. Therefore, it is evident that the resistance is even higher when datasets with larger size images are adopted.

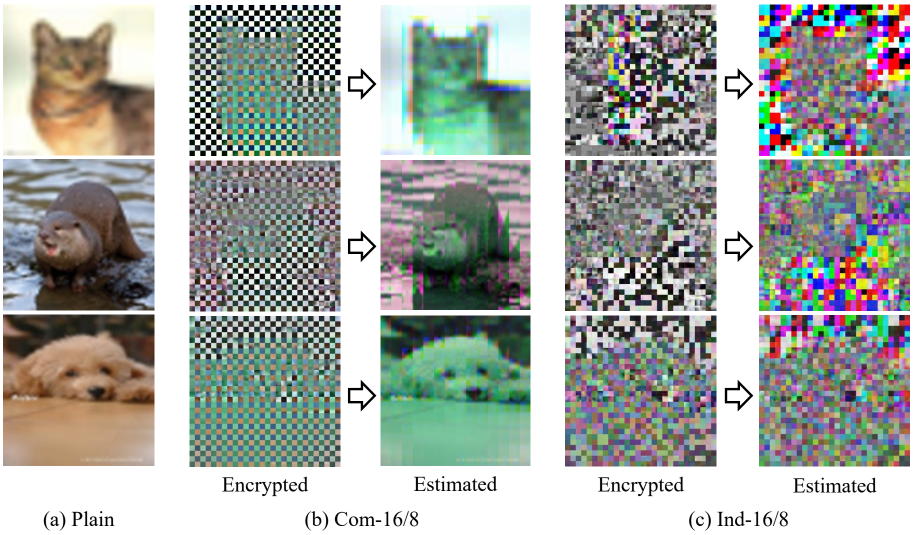


Figure 13. Results of reconstruction by EJPS attack against sub-block-wise EtC images.

5. Conclusions

In this paper, we proposed a novel image encryption method for privacy-preserving ConvMixer using compressible encrypted images. This method allows us not only to assign independent keys to each patch for image encryption but also to maintain the same classification accuracy as that of plain models without encrypted images. In addition, the combined use of the independent key assignment and sub-block wise encryption can enhance robustness against attacks. In image classification experiments, the effectiveness of this method was demonstrated in terms of the accuracy of encrypted models and robustness against attacks including the state-of-the-art attack, EJPS. In this paper, we focused on the use of ConvMixer, but our method is expected to be effective in other isotropic networks such as ViT. We shall evaluate the performance of our method under the use of other networks in our future work.

Author Contributions: Conceptualization, H.L., S.I., and H.K.; methodology, H.L. and H.K.; validation, H.L. and S.I.; investigation, H.L.; writing—original draft preparation, H.L.; writing—review and editing, S.I. and H.K.; supervision, S.I. and H.K.; project administration, S.I. and H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by JSPS KAKENHI, grant number JP21H01327.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kiya, H.; MaungMaung, A.; Kinoshita, Y.; Imaizumi, S.; Shiota, S. An Overview of Compressible and Learnable Image Transformation with Secret Key and Its Applications. *APSIPA Trans. Signal Inf. Process.* **2022**, *11*, e11. [CrossRef]
2. Liu, Y.; Chen, H.; Yang, Z. Enforcing End-to-End Security for Remote Conference Applications. In Proceedings of the 45th IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2024; pp. 2630–2647.
3. Chen, Q.; Wang, Y.; Wang, W.; Nakachi, T.; Zhang, Z. Privacy-Preserving Resource Management for Distributed Collaborative Edge Caching Systems. *IEEE Internet Things J.* **2024**, *11*, 34296–34311. [CrossRef]
4. Yuge, N.; Ishihara, H.; Nakamura, M.; Nakachi, T. Privacy Preserving Deep Unrolling Methods and Its Application to Image Reconstruction. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2024**. [CrossRef]

5. Shimizu, K.; Suzuki, T. Flexibly-Tunable Bitcube-Based Perceptual Encryption within Jpeg Compression. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2702–2706.
6. Hinojosa, C.; Marquez, M.; Arguello, H.; Adeli, E.; Fei-Fei, L.; Niebles, J.C. PrivHAR: Recognizing Human Actions From Privacy-Preserving Lens. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 314–332.
7. Bhaskar, M. *Cryptographic and Information Security Approaches for Images and Videos*, 1st ed.; Cryptographic Image Scrambling Techniques; CRC Press: Boca Raton, FL, USA, 2019; pp. 37–65. ISBN 978-0-429-43546-1.
8. Rohhila, S.; Singh, A.K. Deep Learning-Based Encryption for Secure Transmission Digital Images: A Survey. *Comput. Electr. Eng.* **2024**, *116*, 109236. [CrossRef]
9. Perez, F.; Lopez, J.; Arguello, H. Privacy-Preserving Deep Learning Using Deformable Operators for Secure Task Learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 5980–5984.
10. Sirichotedumrong, W.; Chuman, T.; Imaizumi, S.; Kiya, H. Grayscale-Based Block Scrambling Image Encryption for Social Networking Services. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
11. Ahmad, I.; Choi, W.; Shin, S. Comprehensive Analysis of Compressible Perceptual Encryption Methods—Compression and Encryption Perspectives. *Sensors* **2023**, *23*, 4057. [CrossRef]
12. Tanaka, M. Learnable Image Encryption. In Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taichung, Taiwan, 19–21 May 2018; pp. 1–2.
13. Madono, K.; Tanaka, M.; Onishi, M.; Ogawa, T. Block-Wise Scrambled Image Recognition Using Adaptation Network. In Proceedings of the Workshop on AAAI Conference Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
14. AprilPyone, M.; Kiya, H. Privacy-Preserving Image Classification Using an Isotropic Network. *IEEE Multimed.* **2022**, *29*, 23-33. [CrossRef]
15. Phong, L.T.; Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1333–1345. [CrossRef]
16. Hijazi, N.M.; Aloqaily, M.; Guizani, M.; Ouni, B.; Karray, F. Secure Federated Learning With Fully Homomorphic Encryption for IoT Communications. *IEEE Internet Things J.* **2024**, *11*, 4289–4300. [CrossRef]
17. Lee, J.-W.; Kang, H.; Lee, Y.; Choi, W.; Eom, J.; Deryabin, M.; Lee, E.; Lee, J.; Yoo, D.; Kim, Y.-S.; et al. Privacy-Preserving Machine Learning With Fully Homomorphic Encryption for Deep Neural Network. *IEEE Access* **2022**, *10*, 30039–30054. [CrossRef]
18. Al Badawi, A.; Jin, C.; Lin, J.; Mun, C.F.; Jie, S.J.; Tan, B.H.M.; Nan, X.; Aung, K.M.M.; Chandrasekhar, V.R. Towards the AlexNet Moment for Homomorphic Encryption: HCNN, the First Homomorphic CNN on Encrypted Data With GPUs. *IEEE Trans. Emerging Top. Comput.* **2020**, *9*, 1330–1343. [CrossRef]
19. Ali, A.; Pasha, M.F.; Guerrieri, A.; Guzzo, A.; Sun, X.; Saeed, A.; Hussain, A.; Fortino, G. A Novel Homomorphic Encryption and Consortium Blockchain-Based Hybrid Deep Learning Model for Industrial Internet of Medical Things. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 2402–2418. [CrossRef]
20. Li, L.; Fan, Y.; Tse, M.; Lin, K.-Y. A Review of Applications in Federated Learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [CrossRef]
21. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv* **2016**, arXiv:1610.05492.
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
23. Kiya, H.; Iijima, R.; Aprilpyone, M.; Kinoshita, Y. Image and Model Transformation with Secret Key for Vision Transformer. *IEICE Trans. Inf. Syst.* **2023**, *E106*, 2–11. [CrossRef]
24. Chuman, T.; Kiya, H. A Jigsaw Puzzle Solver-Based Attack on Image Encryption Using Vision Transformer for Privacy-Preserving DNNs. *Information* **2023**, *14*, 311. [CrossRef]
25. Trockman, A.; Kolter, J.Z. Patches are all you need? *arXiv* **2022**, arXiv:2201.09792.
26. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, Department of Computer Science, University of Toronto, Toronto, ON, Canada, 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 23 September 2024).
27. Ishikawa, Y.; Kondo, M.; Kataoka, H. Learnable Cube-Based Video Encryption for Privacy-Preserving Action Recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2024; pp. 7003–7013.
28. Dave, I.R.; Chen, C.; Shah, M. SPAct: Self-Supervised Privacy Preservation for Action Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 20132–20141.
29. Hao, F.; He, F.; Wang, Y.; Wu, F.; Cheng, J.; Tao, D. Privacy-Preserving Vision Transformer on Permutation-Encrypted Images. Available online: https://openreview.net/forum?id=eL1iX7DMnPI (accessed on 4 October 2024).
30. Iijima, R.; Shiota, S.; Kiya, H. A Random Ensemble of Encrypted Vision Transformers for Adversarially Robust Defense. *IEEE Access* **2024**, *12*, 69206–69216. [CrossRef]

31. AprilPyone, M.; Kiya, H. A protection method of trained CNN model with a secret key from unauthorized access. *APSIPA Trans. Signal Inf. Process.* **2021**, *10*, e10.
32. Sirichotedumrong, W.; Kinoshita, Y.; Kiya, H. Pixel-Based Image Encryption Without Key Management for Privacy-Preserving Deep Neural Networks. *IEEE Access* **2019**, *7*, 177844–177855. [CrossRef]
33. Chang, A.H.; Case, B.M. Attacks on Image Encryption Schemes for Privacy-Preserving Deep Neural Networks. *arXiv* **2020**, arXiv:2004.13263.
34. Ito, H.; Kinoshita, Y.; Aprilpyone, M.; Kiya, H. Image to Perturbation: An Image Transformation Network for Generating Visually Protected Images for Privacy-Preserving Deep Neural Networks. *IEEE Access* **2021**, *9*, 64629–64638. [CrossRef]
35. Baidya, R.; Jeong, H. YOLOv5 with ConvMixer Prediction Heads for Precise Object Detection in Drone Imagery. *Sensors* **2022**, *22*, 8424. [CrossRef] [PubMed]
36. Ross, W. Pytorch Image Models. Available online: https://github.com/huggingface/pytorch-image-models (accessed on 23 September 2024).
37. Independent JPEG Group. Available online: https://www.ijg.org (accessed on 23 September 2024).
38. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.