Journal of *Imaging*

MDPI

Article

Privacy-Preserving Semantic Segmentation Using Vision Transformer

Hitoshi Kiya [1,*], Teru Nagamori [1], Shoko Imaizumi [2] and Sayaka Shiota [1]

[1] Department of Computer Science, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino-shi, Tokyo 191-0065, Japan
[2] Graduate School of Engineering, Chiba University, 1-33 Yayoicho, Chiba 263-8522, Japan
* Correspondence: kiya@tmu.ac.jp; Tel.: +81-42-585-8454

Abstract: In this paper, we propose a privacy-preserving semantic segmentation method that uses encrypted images and models with the vision transformer (ViT), called the segmentation transformer (SETR). The combined use of encrypted images and SETR allows us not only to apply images without sensitive visual information to SETR as query images but to also maintain the same accuracy as that of using plain images. Previously, privacy-preserving methods with encrypted images for deep neural networks have focused on image classification tasks. In addition, the conventional methods result in a lower accuracy than models trained with plain images due to the influence of image encryption. To overcome these issues, a novel method for privacy-preserving semantic segmentation is proposed by using an embedding that the ViT structure has for the first time. In experiments, the proposed privacy-preserving semantic segmentation was demonstrated to have the same accuracy as that of using plain images under the use of encrypted images.

Keywords: semantic segmentation; vision transformer; segmentation transformer; privacy-preserving

Citation: Kiya, H.; Nagamori, T.; Imaizumi, S.; Shiota, S. Privacy-Preserving Semantic Segmentation Using Vision Transformer. *J. Imaging* **2022**, *8*, 233. https://doi.org/10.3390/jimaging8090233

Academic Editor: Silvia Liberata Ullo

Received: 14 July 2022
Accepted: 28 August 2022
Published: 30 August 2022

1. Introduction

Deep learning has been deployed in many applications including security-critical ones such as biometric authentication and medical image analysis. Generally, data contains sensitive information, so privacy-preserving methods for deep learning have become an urgent problem. In particular, data including sensitive information cannot be transferred to untrusted third-party cloud environments even if they provide a powerful computing environment. Therefore, it has been challenging to test a deep learning model in cloud environments while preserving privacy. To address the privacy issue, researchers have proposed various solutions. However, cryptographic methods such as fully homomorphic encryption are still computationally expensive [1–3], and moreover, the encrypted images cannot be directly applied to models trained with plain images. To protect privacy, federal learning (FL) has also been studied as a type of distributed machine learning [4,5]. FL is capable of significantly preserving clients' private data from being exposed to adversaries. However, FL aims to construct models over multiple participants without directly sharing their raw data, so the privacy of input (query) images is not considered. For these reasons, numerous learnable perceptual encryption methods have been studied so far for various applications [6–11] that have been inspired by encryption methods for privacy-preserving photo cloud sharing services [12]. Most perceptual image encryption methods aim to realize the secure transmission/storage of images as in [13]. In contrast, learnable image encryption is encryption that allows us not only to generate visually protected images to protect personally identifiable information included in an image such as an individual or the time and location of the taken photograph but to also apply encrypted images to a machine learning algorithm in the encrypted domain.

Perceptual encryption-based methods have been verified to be effective in image classification tasks [7–9,14], but other tasks such as semantic segmentation have never been

considered under the use of perceptual encryption-based methods because such tasks are required to achieve a pixel-level accuracy [15]. Accordingly, in this paper, we propose a novel method for privacy-preserving semantic segmentation that uses encrypted images and models with the vision transformer (ViT) [16], called the segmentation transformer (SETR) [17]. The reason that we focus on ViT–based models is given below. ViT is well known to have a higher performance than conventional convolutional neural networks (CNNs) in some settings. In addition, following the success of ViT in image classification tasks, it is expected to have success in other tasks including semantic segmentation, and ViT also has an embedding structure. In particular, the embedding structure plays an important role in the proposed method. In this paper, we point out for the first time that embedding enables us not only to avoid the influence of block-wise encryption but to also update a secret key easily in a semantic segmentation task, which conventional methods cannot.

We make the following contributions in this paper.

- We propose the combined use of encrypted images and models in a semantic segmentation task to protect visual sensitive information of input images for the first time.
- We confirm that the proposed method allows us not only to use the same accuracy as that when images are not encrypted but to also update a secret key easily.

In addition, the proposed method does not need any network modification. If other models with an embedding structure are developed, the proposed method is also expected to be effective under the use of these models as well.

The rest of this paper consists of related work, the proposed privacy-preserving semantic segmentation, experiments, and a discussion before the conclusion.

2. Related Work

2.1. Privacy-Preserving DNNs

Privacy-preserving machine learning methods with homomorphic encryption (HE) [1,18–22] have been studied. One is CryptoNet [21], which can apply HE to the influence stage of DNNs. CryptoNet has very high computational complexity, so a dedicated low computer convolution core architecture for CryptoNet was proposed and implemented with CMOS technology [22]. In CryptoNet, all activation functions and the loss function must be polynomial functions. Therefore, privacy-preserving machine learning methods with HE are still difficult to be applied to state-of-the-art DNNs.

In comparison, an approach with HE was proposed for privacy-preserving weight transmission for multiple owners who wish to apply a machine learning method over combined data sets [1,18–20]. In this approach, since the gradients are encrypted by using HE, model information is not leaked. The privacy-preserving weight transmission can provide robustness against model extraction attacks. However, this approach does not aim to protect sensitive information of input (query) images. To protect privacy, federal learning (FL) has also been studied as a type of distributed machine learning [4,5]. FL is capable of significantly preserving clients' private data from being exposed to adversaries. Clients store training data locally and use the data to train a local model. Then, the clients upload the trained parameters to a server. In this way, each client can collaboratively train one model on the server protecting the privacy of the data. However, FL aims to construct models over multiple participants without directly sharing their raw data, so the privacy of input images is not considered. The proposed method allows us to protect sensitive information of input images without any performance degradation.

2.2. Learnable Image Encryption for Machine Learning

Learnable encryption encrypts images with a secret key so that visual information in encrypted images is not perceptible to humans while maintaining the ability to classify encrypted images with a model, where a model is trained by using encrypted images. The first concept was introduced for traditional machine learning such as support vector machine (SVM) and random forests [23–25]. For deep learning, Tanaka first introduced block-wise learnable image encryption (LE) with an adaptation layer that is used prior to the classifier to reduce the influence of image encryption [7]. Another encryption method is pixel-wise encryption (PE) in which negative-positive transformation and color component shuffling are applied without using any adaptation layer [9]. However, both block-wise and pixel-wise encryption methods can be attacked by ciphertext-only attacks [6,26]. To enhance the security of encryption, Tanaka's method was extended by adding a block-scrambling step and utilizing different block keys for the pixel encryption operation [8] (hereinafter denoted as ELE). In addition, to improve both the security of encryption and the accuracy of models, the use of isotropic networks such as ViT has been investigated [27,28]. However, there are still several issues: lower accuracy than models trained with plain images, unknown applicability to semantic segmentation, and the lack of an update method for the key. Accordingly, we aim to overcome these issues in this paper.

2.3. Segmentation Transformer

The purpose of semantic segmentation is to classify objects in a pixel-level resolution. Figure 1 shows an overview of semantic segmentation. A segmentation model predicts a segmentation map from an input image, where each pixel in the segmentation map represents a class label. The segmentation transformer (SETR) is the first transformer-based model proposed for semantic segmentation [17]. This model is inspired by ViT, which has a high performance in image classification tasks.

Figure 2 shows the architecture of SETR, where the encoder is the same as in ViT. Since the encoder in ViT receives only 1-D vectors as an input, an image is divided into patches. Then, each patch is flattened and converted to a 1-D vector. Two embeddings are used to understand the location information of separated patches in an original image, and each patch is mapped to learnable dimensions. The former, called position embedding, embeds location information about where each patch is in an original image. The latter, called patch embedding, maps each patch to learnable dimensions using a matrix. By using these two types of embeddings, learning is possible even when an image is divided into patches while keeping location information [27]. After that, the feature representation obtained by the encoder is inputted to the decoder, which outputs a segmentation map of the same size as an input image. The proposed method is carried out on the basis of the embedding structure.

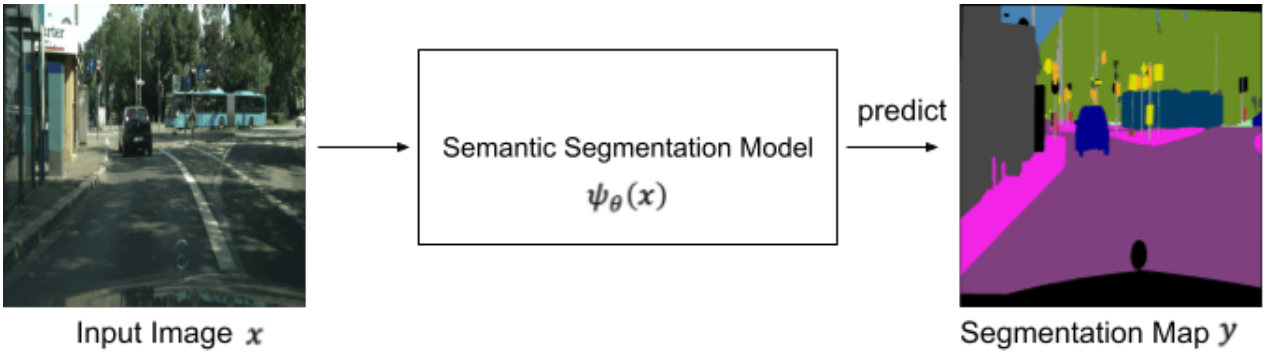

Figure 1. Overview of semantic segmentation.

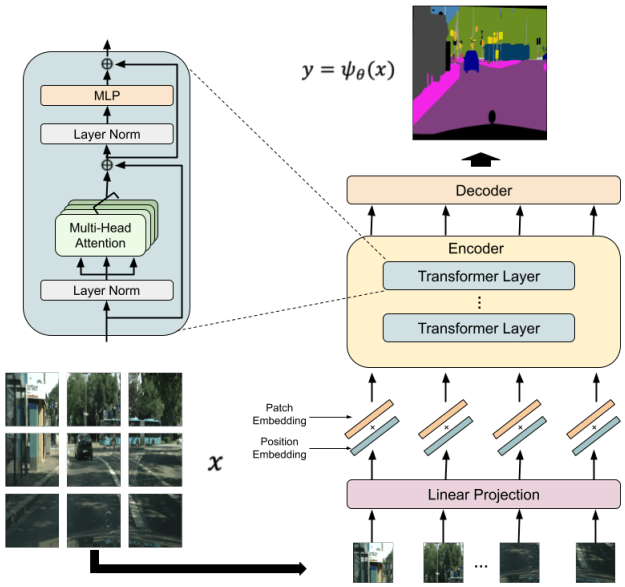

Figure 2. Architecture of segmentation transformer [17].

3. Proposed Method

3.1. Overview and Threat Model

Figure 3 shows an overview of the proposed method for protecting visual information on plain images for semantic segmentation. The method is summarized as below.

First, a model creator trains a model with plain images. Next, after training, the model $\psi_\theta(\cdot)$ is encrypted with a secret key K as $\hat{\psi}_\theta(\cdot)$. Then, key K is provided to authorized users, and the protected model is provided to a provider, who has no key. Then, the users encrypt query (input) images with key K to protect sensitive information included in the images and send the encrypted ones to the server. Finally, the users receive segmentation maps from the server. The main purpose of this work is to protect sensitive information of the input images the users have. The proposed method can be carried out without both any performance degradation and network modification, compared with the use of plain images.

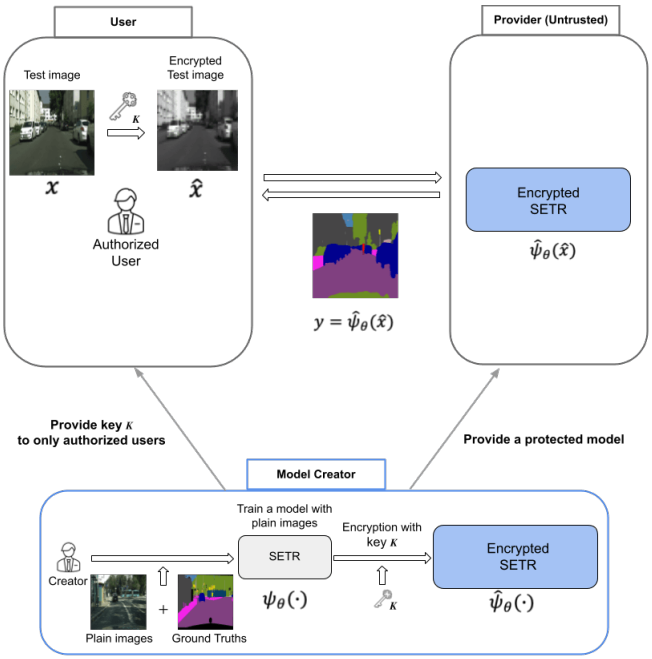

Figure 3. Overview of privacy-preserving semantic segmentation. (SETR: segmentation transformer).

As we focus on protecting personally identifiable information included in an image such as an individual or the time and each car's number in a semantic segmentation scenario, the goal of an adversary is to illegally restore such personally identifiable information. Encrypted images are transferred to an untrusted provider for testing query data as in Figure 3. In this paper, we assume the adversary knows the encryption algorithm but not the secret key. In other words, we assume that the adversary can carry out a ciphertext-only attack (COA) by using only query images that the adversary receives, as in conventional perceptual encryption methods [6].

3.2. Encryption Method

ViT utilizes patch embedding and position embedding. In this paper, we use a unique property of the embedding structure. In SETR, an image tensor $x \in \mathbb{R}^{h \times w \times c}$ is divided into N patches with a size of $p \times p$ where h, w and c denote the height, width, and the number of channels of an image. When h and w are dividable by p, N is given as hw/p^2. After that, each patch is flattened as $x_p^i = [x_p^i(1), x_p^i(2), \ldots, x_p^i(L)]$, and the resulting 1-D vector is linearly mapped to a vector with dimensions of D with a learnable matrix $\mathbf{E}$ as

$$z_0 = [x_{class}; x_p^1\mathbf{E}; x_p^2\mathbf{E}; \ldots x_p^i\mathbf{E}; \ldots x_p^N\mathbf{E}] + \mathbf{E_{pos}}, \tag{1}$$

where

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

x_{class} is the classification token which is the input to MLP (see Figure 2), e_{pos}^0 is the information of the classification token, e_{pos}^i is the position information of each patch, and z_0 is the embedded patch. $\mathbf{E}$ and $\mathbf{E_{pos}}$ are decided by training a model with plain images. By using a trained model ψ_θ, a segmentation map y is given by

$$y = \psi_\theta(x). \tag{2}$$

The proposed method is carried out in accordance with the above relation.

3.2.1. Model Encryption

In the proposed method, trained models have to be encrypted to use encrypted input images. A matrix $\mathbf{E}$ is transformed with key K after training a model as follows.

(1) Randomly generate a matrix $\mathbf{E_{enc}}$ with key K as

$$\mathbf{E_{enc}} = \begin{bmatrix} k_{(1,1)} & k_{(1,2)} & \ldots & k_{(1,L)} \\ k_{(2,1)} & k_{(2,2)} & \ldots & k_{(2,L)} \\ \vdots & \vdots & \ddots & \vdots \\ k_{(L,1)} & k_{(L,2)} & \ldots & k_{(L,L)} \end{bmatrix}, \tag{3}$$

$$\begin{aligned} & k_{(i,j)} \in \mathbb{R}, \ i,j \in \{1, \ldots, L\}, \\ & \mathbf{E_{enc}} \in \mathbb{R}^{L \times L}, \text{and } \det\mathbf{E_{enc}} \neq 0. \end{aligned}$$

(2) Multiply $\mathbf{E_{enc}}$ and $\mathbf{E}$ to obtain $\hat{\mathbf{E}}$ as

$$\hat{\mathbf{E}} = \mathbf{E_{enc}}\mathbf{E}, \ \hat{\mathbf{E}} \in \mathbb{R}^{L \times D}. \tag{4}$$

(3) Replace $\mathbf{E}$ in Equation (1) with $\hat{\mathbf{E}}$ as a new patch embedding to encrypt a model.

Namely, an encrypted model $\hat{\psi}_\theta$ is given by

$$\hat{\psi}_\theta = t(\psi_\theta, \mathbf{E_{enc}}), \tag{5}$$

where $t(\cdot)$ is the proposed model encryption algorithm.

3.2.2. Example of $\mathbf{E_{enc}}$

There is a lot of flexibility in the design of $\mathbf{E_{enc}}$. A design example of $\mathbf{E_{enc}}$ is given as follows.

(1) Generate a random integer vector with a length of L by using a random generator with a seed value as

$$l_{enc} = [l_e(1), l_e(2), \ldots, l_e(i), \ldots, l_e(L)], \tag{6}$$

where

$$le(i) \in \{1, 2, ..., L\},$$
$$le(i) \neq le(j) \text{ if } i \neq j.$$

(2) Decide $k_{(i,j)}$ in Equation (3) with l_{enc} as

$$k_{(i,j)} = \begin{cases} 0 & (j \neq l_e(i)) \\ 1 & (j = l_e(i)) \end{cases}. \tag{7}$$

For example, if L = 4 and l_{enc} = (4,1,3,2), $\mathbf{E_{enc}}$ is given by

$$\mathbf{E_{enc}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{8}$$

In addition, $\mathbf{E_{enc}}$ in Equation (8) is an orthogonal matrix, so the relation,

$$\mathbf{E_{enc}^{-1}} = \mathbf{E_{enc}^{\top}}, \tag{9}$$

is satisfied where $^\top$ is a transposition. Key K corresponds to a seed value used in a random integer generator.

3.2.3. Test Image Encryption

We assume that an authorized user has key K, so the user can generate $\mathbf{E_{enc}}$ by using K as well. Accordingly, a new matrix $\hat{\mathbf{E}}_{\mathbf{enc}}$ can be produced by

$$\hat{\mathbf{E}}_{\mathbf{enc}} = \mathbf{E_{enc}^{-1}}. \tag{10}$$

An encrypted test image $\hat{x} \in \mathbb{R}^{h \times w \times c}$ is produced by an authorized user as follows.

(a) Divide a test (query) image tensor $x \in \mathbb{R}^{h \times w \times c}$ into blocks with a size of $p \times p$ such that $B = \{B_1, \ldots, B_N\}$.

(b) Flatten each block B_i into a vector b_i such that

$$b_i = [b_i(1), \ldots, b_i(L)], \tag{11}$$

(c) Generate an encrypted vector $\hat{b}_i$ by multiplying b_i by $\hat{\mathbf{E}}_{\mathbf{enc}}$ as

$$\hat{b}_i = b_i \hat{\mathbf{E}}_{\mathbf{enc}}, \ \hat{b}_i \in \mathbb{R}^L, \tag{12}$$

(d) Concatenate the encrypted vectors into an encrypted test image $\hat{x}$.

As a result, when $\hat{x}$ is applied to the encrypted model, the embedded patch becomes

$$\begin{aligned} z_0 =& [x_{class}; \hat{x}_p^1\hat{\mathbf{E}}; \hat{x}_p^2\hat{\mathbf{E}}; \dots \hat{x}_p^i\hat{\mathbf{E}}; \dots \hat{x}_p^N\hat{\mathbf{E}}] + \mathbf{E_{pos}}, \\ =& [x_{class}; x_p^1\mathbf{E}; x_p^2\mathbf{E}; \dots x_p^i\mathbf{E}; \dots x_p^N\mathbf{E}] + \mathbf{E_{pos}}, \end{aligned} \tag{13}$$

where $\hat{x}_p^i$ is the i-th patch of an encrypted test image.

Accordingly, when $\hat{x}$ is inputted to an encrypted model $\hat{\psi}_\theta$, an output $\hat{y}$ is given by

$$\begin{aligned} \hat{y} =& \hat{\psi}_\theta(\hat{x}) \\ =& \psi_\theta(x) = y. \end{aligned} \tag{14}$$

From Equation (14), the combined use of encrypted image $\hat{x}$ and encrypted model $\hat{\psi}_\theta$ can give the same output as that of plain model ψ_θ.

When Equations (7) and (8) are chosen as $\mathbf{E_{enc}}$, $\hat{\mathbf{E}}_{\mathbf{enc}}$ is given by

$$\hat{\mathbf{E}}_{\mathbf{enc}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

This selection corresponds to pixel shuffling (SHF) in each block.

3.3. Requirements of Proposed Method

The proposed method satisfies the following requirements.

(a) Semantic segmentation can be carried out by using visually protected input images without sensitive information.
(b) No network modification is required.
(c) A high accuracy, which is close to that of using plain images, can be maintained.
(d) Keys are easily updated.

Requirement (a) is the main purpose of this work. As described in Section 3.2, the proposed method encrypts trained models on the basis of a matrix transformation, so there is no need to modify the structure of models. In addition, under Equation (10), the combined use of the encrypted images and model can produce the same result as that without any encryption. When we want to update key K, the model is easily updated by using a new $\mathbf{E_{enc}}$ generated with a new key. Accordingly, the proposed method can satisfy all the above requirements.

Although many privacy-preserving DNNs have been studied so for, some of them focus on privacy-preserving model training [1,4,5,18–20]. Numerous methods can protect the visual information of input images, but the methods are not capable of semantic segmentation tasks [6–12,14]. In addition, they degrade the performance of models compared with that without any encryption. Accordingly, the proposed method enables to satisfy all the above requirements for the first time.

4. Experimental Results

4.1. Setup

We conducted semantic segmentation experiments on two datasets to verify the effectiveness of the proposed method. The first dataset was Cityscapes [29], which is an urban scene dataset with 19 object categories. It consists of 5000 images with a resolution of 2048 × 1024 in total, and the images were divided into 2975, 500, and 1525 sets for training, validation, and testing, respectively. The other dataset was ADE20K [30], which is a benchmark for scene analysis with 150 categories. It consists of 20,210, 2000, and 3352 images for training, validation, and testing, respectively. In the experiments, training and validation images were used for training and testing, respectively. These two datasets used in this experiment are the same as in the paper [17] in which SETR was proposed. In

this paper, the effectiveness of our method was confirmed by using these datasets. The effectiveness of the method does not depend on datasets.

In SETR [17], the encoder has two variations: T-base and T-large. T-base is a small model with 12 transformer layers and 768 hidden layers, and T-large is a large model with 24 transformer layers and 1024 hidden layers. In addition, there are three types of decoders: *Naïve*, which employs a simple two-layer network followed by a bilinear upsampler to restore an original resolution, *PUP*, which alternates between convolution layers and upsampling operations, and *MLA*, which uses multi-stage features such as a feature pyramid network.

In the experiment, all types of decoders were used on the two datasets. In addition, for Cityscapes, T-base was used as the encoder, and pre-trained weights of Deit [31] were utilized to initialize all transformer layers and input the linear projection layers of the model. For ADE20K, T-large was used as the encoder, and the pre-trained weights of ViT were utilized to initialize all transformer layers and input the linear projection layers of the model. In addition, input images were resized to images with a size of 768×768 for Cityscapes and to images with a size of 512×512 for ADE20K. Other training conditions including data augmentation methods used in the experiments were the same as those in [17].

As an evaluation metric, we used mean intersection-over-union ($mIoU$), which is an average of the intersection-over-union (IoU) for each class defined as

$$IoU = \frac{TP}{TP + FP + FN}, \tag{16}$$

where TP, FP, and FN mean true positive, false positive, and false negative values calculated from a predicted full-resolution output and ground truth, respectively. When a IoU value is closer to 1, it indicates a higher accuracy.

4.2. Semantic Segmentation Performance

In the experiment, SETR models with a patch size of $p = 16$ were trained by using plain images, and then trained models were encrypted with a secret key K in accordance with the proposed procedure, where $\mathbf{E}_{enc}$ was generated on the basis of Equations (6) and (7). In addition, test images were encrypted with key K as well. Figure 4 shows an example of encrypted images. From the figure, sensitive visual information in images was confirmed to be protected by using image encryption. Table 1 also shows experimental results for each decoder for the two datasets where baseline represents the results for the models without encryption, and Correct(K) represents the results when an authorized user applied images encrypted with key K to the encrypted models. From the table, the proposed method allows authorized users not only to protect sensitive information but to also gain the same performance from encrypted models as that from plain models.

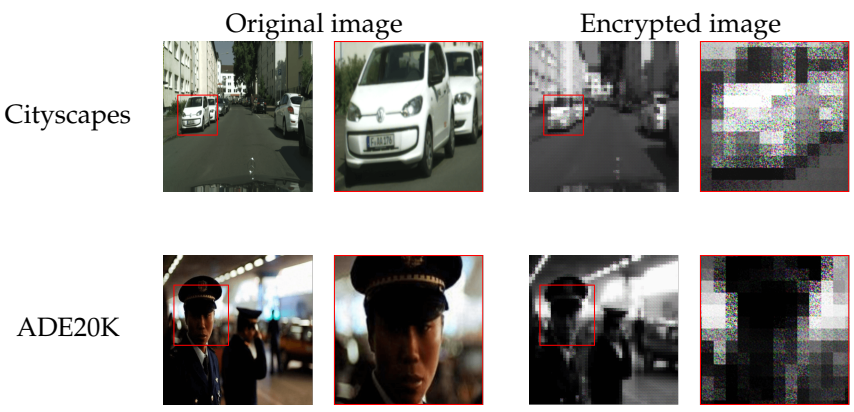

Figure 4. Example of encrypted images ($p \times p = 16 \times 16$). Zoom-ins of red-framed regions are shown on right side of each image. The red boxes represent sensitive information such as license plates.

Random (K') and No-enc in Table 1 show the results for unauthorized users. In the experiment, we randomly generated 50 keys for Random (K'), and the average value of 50 trials was computed under each condition. In addition, plain images were applied to encrypted images for No-enc. From the table, unauthorized users without key K could not gain high performance from the encrypted models. An example of predicted segmentation maps is given in Figure 5. From the figure, the effectiveness of the method was also verified. Figure 6 also shows the performance of the models in more detail when randomly generated 50 keys were used. The highest value of mIoU in the experiment was 0.13, which was still low.

Table 1. Accuracy of proposed models (*mIoU*).

Dataset	Selected Decoder	Baseline	Correct (K)	No-Enc	Random (K')
Cityscapes	*Naïve*	0.6490	0.6490	0.0674	0.0718
	MLA	0.6386	0.6386	0.0792	0.0743
	PUP	0.7039	0.7039	0.1135	0.1137
ADE20K	*Naïve*	0.3710	0.3710	0.0023	0.0024
	MLA	0.4370	0.4370	0.0030	0.0029
	PUP	0.4383	0.4383	0.0048	0.0050

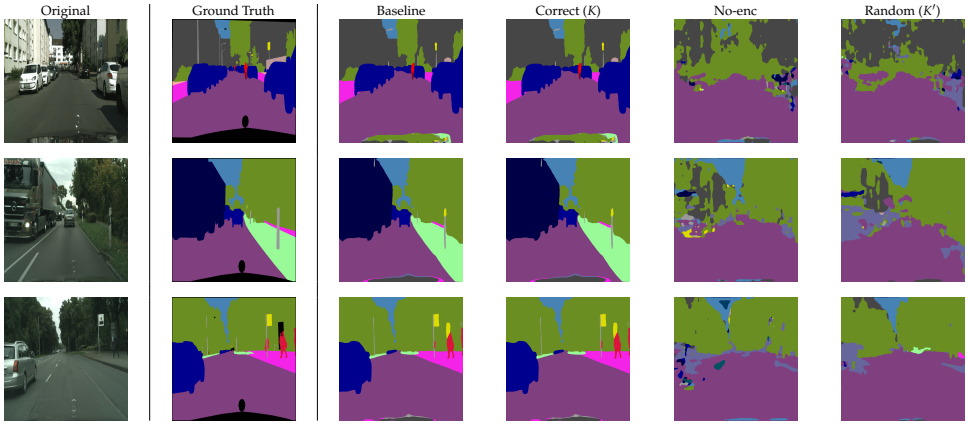

Figure 5. Example of predicted segmentation maps (with PUP on Cityscapes).

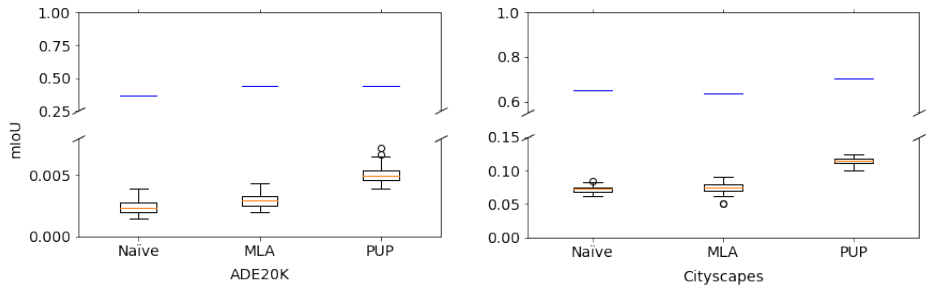

Figure 6. Mean IoU (*mIoU*) values of protected models with randomly generated 50 keys. Boxes span from first to third quartile, referred to as Q_1 and Q_3, and whiskers show maximum and minimum values in range of $[Q_1 - 1.5(Q_3 - Q_1), Q_3 + 1.5(Q_3 - Q_1)]$. Band inside box indicates median. Outliers are indicated as dots. Blue lines represent each baseline.

4.3. Comparison with Conventional Methods

In this paper, we proposed a privacy-preserving semantic segmentation method to protect visually sensitive information of input images for the first time. Many conventional methods for privacy-preserving DNNs do not consider protecting sensitive information of input images such as federal learning [1,4,5,18–20]. Several methods can protect sensitive information of input images, in which encrypted images are used for model training, but

the use of encrypted images for model training is known to degrade the performance of models [6–12,14]. In particular, for privacy-preserving semantic segmentation, the performance of models is heavily degraded in general because a pixel-level resolution is required for semantic segmentation [15]. In addition, when updating the key, the model has to be retrained by using images encrypted with a new key.

To compare the proposed method with conventional methods, Table 2 shows the results of CNN models trained with images encrypted by a block-wise encryption method, which was proposed in [32]. In [32], three encryption methods: pixel shuffling (SHF), negative/positive transformation (NP), and format-preserving Feistel-based encryption (FFX) were proposed. From the table, encrypting images to protect sensitive information in CNNs without embedding structures significantly degraded the accuracy of models compared to baseline due to the influence of encryption, even if the correct key was used. This problem is caused by the collapse of spatial information because CNNs do not have structures to store pixel location information. In contrast, the proposed method focuses on a transformer model that has embedding structures to preserve positional information, and thus the proposed method can maintain accuracy while protecting sensitive information in images.

Table 2. Accuracy (mIoU) of conventional method with encrypted images [15].

Network	**Fully Convolutional Network (FCN)**								
Block size	SHF			NP			FFX		
	Correct (K)	No-enc	Random (K')	Correct (K)	No-enc	Random (K')	Correct (K)	No-enc	Random (K')
4	0.4731	0.4536	0.3671	0.4706	0.3359	0.1505	0.3823	0.0157	0.0012
16	0.2214	0.1994	0.1150	0.3439	0.2114	0.0832	0.2611	0.0007	0.0079
Baseline	0.5966								

4.4. Robustness against Attacks

Encrypted images have to be robust against attacks that aim to restore sensitive visual information from encrypted images. Numerous attack methods have been proposed to evaluate the robustness of perceptual encryption methods [26,33–36]. Existing learnable encryption methods have been evaluated under ciphertext-only attacks (COA). In this paper, to evaluate the robustness of encrypted images used in SETR, we considered brute-force attacks and the feature reconstruction attack (FR-Attack) [26], which exploits the local properties of an encrypted image to reconstruct visual information from encrypted images as a COA. Furthermore, as the distribution of the dataset is known, we also considered that the adversary may prepare exact pairs of plain images and encrypted ones with multiple different keys to learn a transformation model, i.e., the inverse transformation network attack (ITN-Attack) [34]. In addition, since images are encrypted by using a block-wise method, a jigsaw puzzle solver attack [35,36] was also used for evaluating the robustness of the proposed method.

(1) Key Space: The key space describes a set of all possible keys in an encryption algorithm. For the case where an image is divided into blocks with a size of $p \times p$, the key space of the proposed algorithm (pixel shuffling) in Equation (7) is given as below.

$$S_{key} = (p \times p \times c)! \tag{17}$$

For example, when $p \times p$ is chosen as the patch size of SETR, the key space is given by

$$S_{key} = (16 \times 16 \times 3)! \simeq 2^{6259}. \tag{18}$$

The use of a large key space enhances robustness against blue-force attacks. Typical cipher systems are recommended to have 2^{128} as a key space as in [37], so the proposed method has a large key space. Accordingly, it is expected to be robust against brute-force attacks.

(2) Robustness Against Attack Methods: Three state-of-the-art attack methods, FR-attack, ITN-attack, and jigsaw puzzle solver attack, were used for evaluation. Figure 7 shows images restored by using the three attacks. Peak signal-to-noise ratio (PSNR) values are marked at the bottom of the restored images to illustrate the perceived sensitivity of the noise component between a restored image and an original one. A larger value means less degradation between the two images. The results from Figure 7 indicate that the encrypted images with the proposed method did not have personally identifiable visual information in the original images even after the attacks. In addition, we also confirmed that the restored images for other test images in the test set followed a similar trend as in Figure 7. Therefore, the proposed method was robust against such attacks.

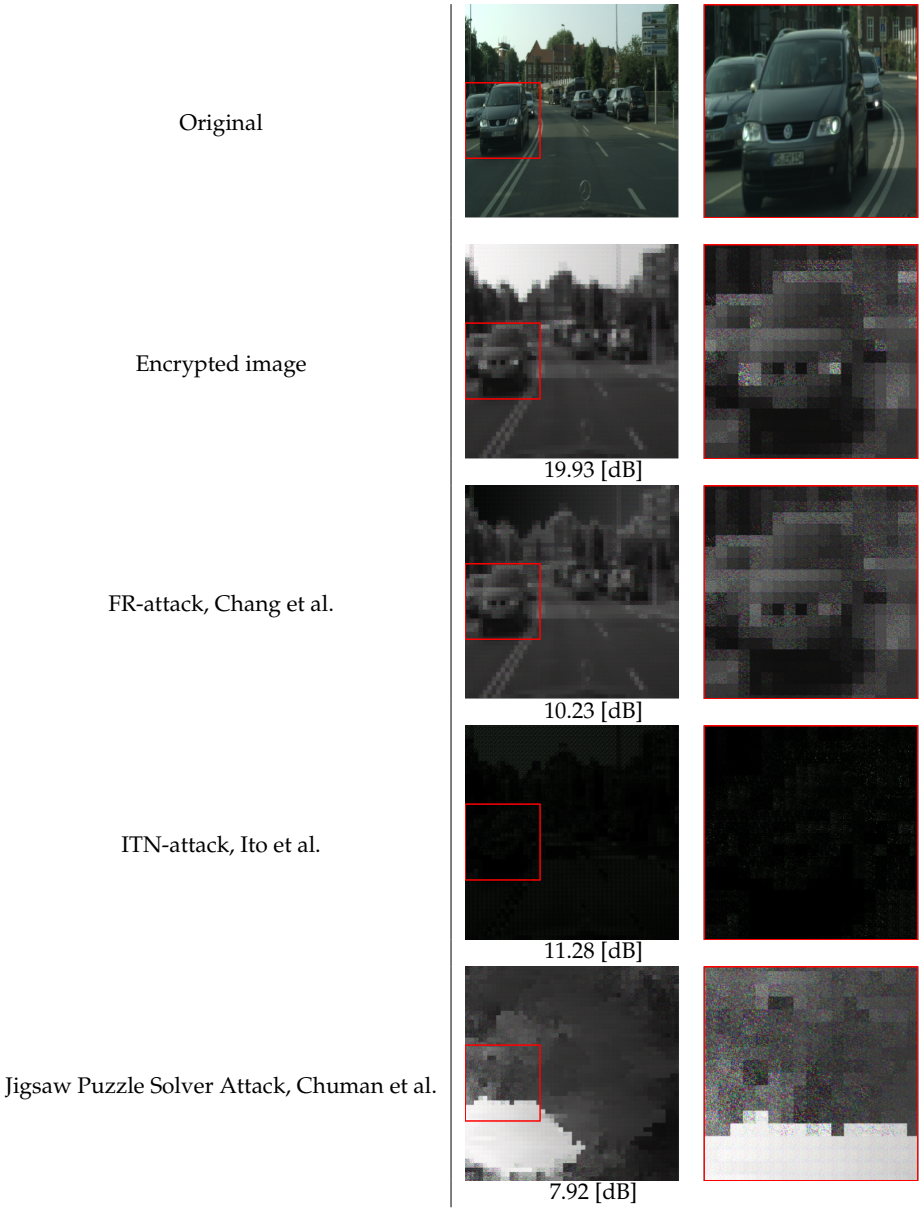

Figure 7. Examples of images restored from encrypted ones, see [26,34–36]. Zoom-ins of red-framed regions are shown on right side of each image. The red boxes represent sensitive information such as license plates. PSNR values are given under images.

As another interesting attack, attackers may learn the transformation matrix between plain images and encrypted images to build a neural network to predict encrypted images given plain images, where attackers do not have the pairs of plain images and images

encrypted with the correct key, so encrypted images have to be predicted by using randomly generated keys. Encrypted images can be predicted from plain images, but the images are different from images encrypted with the correct key. Thus, if these predicted images are applied to encrypted SETR, the accuracy of the estimated segmentation maps will be low as well as for random (K') in Table 1 and Figure 6. We carried out a preliminary experiment to confirm the validity of the above consideration. In addition, this attack cannot restore sensitive visual information from predicted images.

5. Conclusions

In this paper, we proposed the combined use of SETR, which is based on the vision transformer, and encrypted images for privacy-preserving semantic segmentation for the first time. The proposed method is carried out on the basis of the embedding structure that ViT has so that it enables us not only to protect sensitive visual information in plain images but to also use the same accuracy from encrypted models as that of models without encryption. Moreover, it does not need any network modification for privacy-preserving DNNs, and it is possible to easily update the key used for model encryption. In experiments, the proposed method was demonstrated to be effective in terms of segmentation accuracy and robustness against various attacks.

In this paper, the effectiveness of the proposed method was verified under the use of SETR, but the proposed method would be effective under the use of other models with an embedded structure. If models with an embedded structure that perform better than SETR are developed, the method is expected to get a higher performance under the use of these models while protecting sensitive information. As for future work, we shall investigate whether the proposed method can be extended to other networks with an embedding structure.

Author Contributions: Conceptualization, H.K. and T.N; methodology, H.K and T.N.; software, T.N.; validation, H.K. and T.N.; formal analysis, H.K. and T.N.; investigation, H.K., T.N., S.I. and S.S.; resources, H.K.; data curation, H.K and T.N.; writing—original draft preparation, H.K.; writing—review and editing, H.K.; visualization, T.N.; supervision, H.K.; project administration, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partially supported by JSPS KAKENHI (Grant Number JP21H01327) and the Support Center for Advanced Telecommunications Technology Research, Foundation (SCAT).

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 conflict of interest.

References

1. Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Montecello, IL, USA, 29 September–2 October 2015; pp. 909–910. [CrossRef]
2. Chaudhari, H.; Rachuri, R.; Suresh, A. Trident: Efficient 4PC Framework for Privacy Preserving Machine Learning. In Proceedings of the 27th Annual Network and Distributed System Security Symposium (NDSS) 2020, San Diego, CA, USA, 23–26 February 2020. [CrossRef]
3. Kitai, H.; Cruz, J.P.; Yanai, N.; Nishida, N.; Oba, T.; Unagami, Y.; Teruya, T.; Attrapadung, N.; Matsuda, T.; Hanaoka, G. MOBIUS: Model-Oblivious Binarized Neural Networks. *IEEE Access* **2019**, *7*, 139021–139034. [CrossRef]
4. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Poor, H.V. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
5. Zhao, J.; Zhu, H.; Wang, F.; Lu, R.; Liu, Z.; Li, H. PVD-FL: A Privacy-Preserving and Verifiable Decentralized Federated Learning Framework. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2059–2073. [CrossRef]
6. Kiya, H.; AprilPyone, M.; 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]

7. Tanaka, M. Learnable Image Encryption. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taichung, Taiwan, 19–21 May 2018; pp. 1–2. [CrossRef]
8. Madono, K.; Tanaka, M.; Masaki, O.; Tetsuji, O. Block-wise Scrambled Image Recognition Using Adaptation Network. In Proceedings of the Workshop on Artificial Intelligence of Things (AAAI WS), New York, NY, USA, 7–8 February 2020.
9. 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]
10. AprilPyone, M.; Kiya, H. Block-wise Image Transformation with Secret Key for Adversarially Robust Defense. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2709–2723.
11. Huang, Q.X.; Yap, W.L.; Chiu, M.Y.; Sun, H.M. Privacy-Preserving Deep Learning With Learnable Image Encryption on Medical Images. *IEEE Access* **2022**, *10*, 66345–66355. [CrossRef]
12. Chuman, T.; Sirichotedumrong, W.; Kiya, H. Encryption-Then-Compression Systems Using Grayscale-Based Image Encryption for JPEG Images. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1515–1525. [CrossRef]
13. Man, Z.; Li, J.; Di, X.; Sheng, Y.; Liu, Z. Double image encryption algorithm based on neural network and chaos. *Chaos Solitons Fractals* **2021**, *152*, 111318. [CrossRef]
14. Sirichotedumrong, W.; Kiya, H. A GAN-Based Image Transformation Scheme for Privacy-Preserving Deep Neural Networks. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Virtual, 18–22 January 2021; pp. 745–749. [CrossRef]
15. Ito, H.; AprilPyone, M.; Kiya, H. Access Control Using Spatially Invariant Permutation of Feature Maps for Semantic Segmentation Models. In Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 14–17 December 2021; pp. 1833–1838.
16. 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. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, 3–7 May 2021.
17. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 6877–6886. [CrossRef]
18. Aono, Y.; Hayashi, T.; Phong, L.T.; Wang, L. Privacy-preserving logistic regression with distributed data sources via homomorphic encryption. *IEICE Trans. Inf. Syst.* **2016**, *99*, 2079–2089.
19. Aono, Y.; Hayashi, T.; Wang, L.; Moriai, S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 1333–1345.
20. Phuong, T.T. Privacy-preserving deep learning via weight transmission. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 3003–3015.
21. Dowlin, N.; Gilad-Bachrach, R.; Laine, K.; Lauter, K.; Naehrig, M.; Wernsing, J. *CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy*; Technical Report MSR-TR-2016-3; Microsoft Research: Redmond, WA, USA, February 2016.
22. Wang, Y.; Lin, J.; Wang, Z. An efficient convolution core architecture for privacy-preserving deep learning. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
23. Maekawa, T.; Kawamura, A.; Nakachi, T.; Kiya, H. Privacy-Preserving Support Vector Machine Computing Using Random Unitary Transformation. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2019**, *102*, 1849–1855. [CrossRef]
24. Kawamura, A.; Kinoshita, Y.; Nakachi, T.; Shiota, S.; Kiya, H. A Privacy-Preserving Machine Learning Scheme Using EtC Images. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2020**, *103*, 1571–1578. [CrossRef]
25. Nakamura, I.; Tonomura, Y.; Kiya, H. Unitary Transform-Based Template Protection and Its Application to l^2-norm Minimization Problems. *IEICE Trans. Inf. Syst.* **2016**, *99*, 60–68. [CrossRef]
26. Chang, A.H.; Case, B.M. Attacks on Image Encryption Schemes for Privacy-Preserving Deep Neural Networks. *arXiv* **2020**, arXiv:2004.13263.
27. AprilPyone, M.; Kiya, H. Privacy-Preserving Image Classification Using an Isotropic Network. *IEEE MultiMedia* **2022**, *29*, 23–33. [CrossRef]
28. Qi, Z.; AprilPyone, M.; Kinoshita, Y.; Kiya, H. Privacy-Preserving Image Classification Using Vision Transformer. *arXiv* **2022**, arXiv:2205.12041. [CrossRef]
29. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223. [CrossRef]
30. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Semantic Understanding of Scenes Through the ADE20K Dataset. *Int. J. Comput. Vis.* **2018**, *127*, 302–321.
31. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the 38th International Conference on Machine Learning, PMLR, Proceedings of Machine Learning Research, Virtual, 18–24 July 2021; Volume 139, pp. 10347–10357.
32. 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. [CrossRef]

33. Madono, K.; Tanaka, M.; Onishi, M.; Ogawa, T. SIA-GAN: Scrambling Inversion Attack Using Generative Adversarial Network. *IEEE Access* **2021**, *9*, 129385–129393. [CrossRef]
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. Chuman, T.; Kurihara, K.; Kiya, H. On the security of block scrambling-based ETC systems against jigsaw puzzle solver attacks. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2157–2161. [CrossRef]
36. Chuman, T.; Kurihara, K.; Kiya, H. On the Security of Block Scrambling-Based EtC Systems against Extended Jigsaw Puzzle Solver Attacks. *IEICE Trans. Inf. Syst.* **2018**, *101*, 37–44. [CrossRef]
37. Schneier, B.; Sutherland, P. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1995.