electronics

MDPI

Article

Selective Feature Anonymization for Privacy-Preserving Image Data Publishing

Taehoon Kim and Jihoon Yang *

Machine Learning Research Laboratory, Department of Computer Science and Engineering, Sogang University, Seoul 04107, Korea; taehoonkim@sogang.ac.kr

* Correspondence: yangjh@sogang.ac.kr

Received: 3 May 2020; Accepted: 20 May 2020; Published: 25 May 2020

Abstract: There is a strong positive correlation between the development of deep learning and the amount of public data available. Not all data can be released in their raw form because of the risk to the privacy of the related individuals. The main objective of privacy-preserving data publication is to anonymize the data while maintaining their utility. In this paper, we propose a privacy-preserving semi-generative adversarial network (PPSGAN) that selectively adds noise to class-independent features of each image to enable the processed image to maintain its original class label. Our experiments on training classifiers with synthetic datasets anonymized with various methods confirm that PPSGAN shows better utility than other conventional methods, including blurring, noise-adding, filtering, and generation using GANs.

Keywords: adversarial learning; data privacy; deep learning; differential privacy; generative adversarial networks; machine learning; model inversion attacks

1. Introduction

The publication of various benchmark datasets enabled the emergence of a variety of current state-of-the-art deep learning models. However, excellent model performance does not always guarantee the possibility of generalization. Transfer learning and optimization techniques, such as few-shot [1], one-shot [2], and zero-shot [3] learning, might be able to bridge the gap between two different datasets, but none of these can be an optimal solution.

A promising approach for finding a model that works best on a specific data distribution is to train the model with a training set directly sampled from that distribution. The importance of the dataset increases the imbalance between those who possess it and those who do not, which increases individual researchers' reliance on benchmark datasets. The lack of public data makes it difficult for data holders to take advantage of the current open-source flow in the deep learning community.

Although data publication can be beneficial for both data holders and individual researchers, not all data can be published freely in its raw form because of privacy issues. Datasets, including collections of images, speech, or videos, from millions of individuals are ripe with privacy risks. Without the data provider's full consent to publication, the dataset should be either noised with an appropriate level of anonymity or substituted with a synthetic neighboring dataset that has a distribution similar to the original.

Synthetic data generation [4–6] is a technique wherein sensitive data is partially or fully replaced with synthetic data before it is published. Along with recent advancements in generative adversarial networks (GANs) [7–12], synthetic data generation has become the focus in recent years as a fundamental solution for privacy-preserving data publication. Beaulieu-Jones et al. [4] generate shareable biomedical data by applying an objective perturbation [13] to ACGAN [9].

Autoencoders are also widely used as a tool to convert an image into another synthetic image. Ma et al. [14] and Ren et al. [15] manually extract a pose or an action from an image and feed it to the autoencoder with the original to generate a new image with the same pose or action. Kim and Yang [16] anonymize an image by applying Laplace and Gaussian mechanisms [17] to the latent-space-level feature representation of the image and reconstructing it to the original pixel-level. Kim and Yang's approach utilizes substantial privacy-preserving aspects of differential privacy [17] but fails to preserve the original class of the input image because of the indiscriminate noise-adding technique to the image's feature representation.

Conventional image anonymization methods add noise to images at the pixel-level. Modifying an image at the pixel-level is simple and computationally efficient. However, it drastically decreases in utility if a significant level of privacy is applied. The idea of encoding an image at the latent-space-level allows feature manipulation rather than pixel manipulation, increasing the utility of the final result. If it is possible to add noise only to the class-independent features, we can use the processed image data for much broader research topics such as classification, anomaly detection, and data augmentation. Throughout this paper, we define class-dependent features as features common to images in the same class and class-independent features as features unique to each image.

In this work, we propose a privacy-preserving semi-generative adversarial network (PPSGAN) as a novel solution to selective feature anonymization for private data publication. The main contributions of our work focus on the improvement of PPAPNet [16] to enhance the utility of the processed image data as follows:

- We introduce PPSGAN, an image anonymization deep neural network that preserves the privacy of individuals related to the image dataset without losing the usefulness of the entire dataset.
- We use the self-attention mechanism [18] to make the noise amplifier of PPSGAN apply different levels of privacy according to the importance of the feature. This mechanism allows PPSGAN to keep the original class label of each image, even in strict privacy conditions.
- We evaluate the quality and the utility of the image data anonymized with our model from different aspects, including the performance of the classifiers trained with the original data, processed with PPSGAN, and generated or modified with other methods.

PPSGAN consists of two sets of networks: a set of encoder and decoder networks with a noise amplifier and a set of critic and classifier networks. The encoder and decoder networks add noise to class-independent features of the input image, and the critic and classifier networks evaluate the processed image via comparison with real samples. We train two sets in an adversarial setting, a common strategy for training GANs. The encoder converts an image into its latent-space representation, z, a vector that contains the essential features of the image. Unlike PPAPNet, we attach a self-attention module [18] after the encoder to distinguish class-dependent features from z. The noise amplifier references the attention matrix inside the self-attention module to set class-independent features as targets for noise-adding. The decoder reconstructs the modified latent-space representation, $\tilde{z}$, into a new image. To ensure that the decoder is not merely generating random images, we add a penalty to the training loss of the encoder and the decoder. The critic decides whether the processed image is real or fake, and the classifier decides the class. Figure 1 contains a detailed visualization of the model architecture of PPSGAN.

Unlike PPAPNet, we use the ACGAN [9] critic instead of the WGAN-GP [19] critic to guide the training of the self-attention module. The ACGAN critic has an auxiliary classifier that determines whether the image still has its class-dependent features. With feedback from the critic and the classifier, the self-attention module updates its attention matrix for improved discrimination between class-dependent and independent features. While Kim and Yang [16] penalize their model with an attacker, a network that tries to reconstruct the original image from the processed image, we use a simple penalty term, zero-noise penalty. In Figure 2, PPSGAN successfully converts images into visually different images in the same class without the attacker.

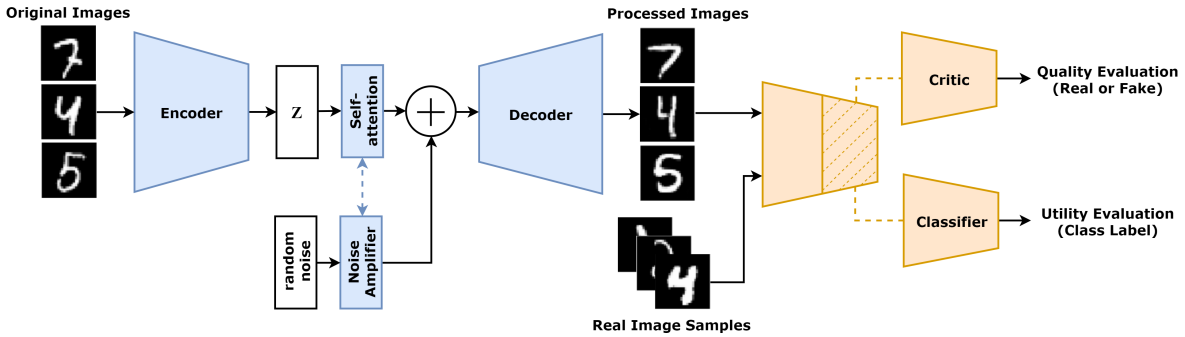


Figure 1. The model architecture of PPSGAN. The encoder–decoder network performs selective feature anonymization to original images. The self-attention module selects class-dependent features. The noise amplifier references the attention matrix of the self-attention module to add noise to class-independent features. The critic and the classifier evaluate processed images. The critic measures the quality (whether it is real or fake), and the classifier measures the utility (whether it preserves the original class label).

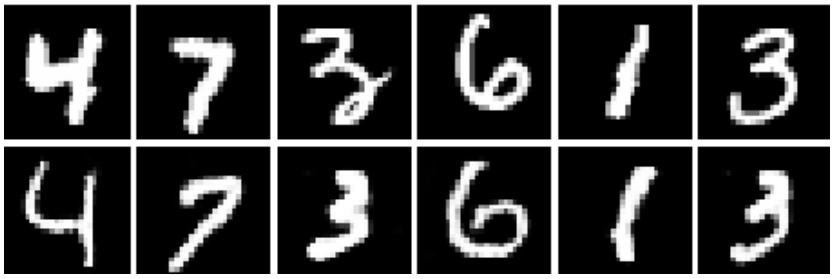


Figure 2. Samples from PPSGAN trained on the MNIST dataset. Our model successfully converts each image (top) into a new image (bottom) while preserving class-dependent features. With our novel selective feature anonymization mechanism, PPSGAN-processed image data can be used for much broader research topics such as classification, anomaly detection, and data augmentation.

2. Background

In this section, we summarize the essential concepts of generative adversarial networks, differential privacy, and self-attention. PPSGAN utilizes the ACGAN [9] critic to perform the quality and utility evaluation of processed images. We fuse differential privacy and self-attention for selective feature anonymization.

2.1. Generative Adversarial Networks

In recent years, generative adversarial networks (GANs) [8–12] have played a pivotal role in the area of data generation and style transfer. The underlying idea is a two-player minimax game between a generator and a critic (discriminator) that trains two networks in an adversarial mode. This methodology minimizes a particular f-divergence between the model distribution (P_θ) and the real distribution (P_r) [20]. Choosing an appropriate f-divergence is essential in preventing a mode collapse, which is a well-known problem when the GAN's generator only draws one or a few foolish examples.The earth mover (EM) distance is one of the most popular f-divergences used in state-of-the-art GANs [7,19,21].

Making slight modifications to the original GAN structure can broaden its usefulness. By replacing the generator with deep convolutional encoder–decoder networks, researchers also perform style transfer with GANs [22–25]. Kim et al. [22] use deep convolutional encoder–decoder networks and a DCGAN [11] critic to find mappings between two different image domains. Odena et al. [9] add an auxiliary classifier to the critic and feed the generator with random noise and a target class label to make images in the target class.

The power of GAN comes from its ability to generate images from random noise that are indistinguishable from real images. If the utility of generated images is guaranteed, we can also replace the original dataset with a synthetic dataset generated with GANs for privacy-preserving purposes. Ren et al. [15] propose a video anonymizer that modifies each person's face with minimal effect on the action detection performance. Ma et al. [14] manipulate the foreground, background, and pose of the input image using different embedding vectors. To preserve the privacy of an individual in an image, Sun et al. [26] replace the face of the target with a randomly generated face image.

While other researchers manually replace and preserve certain features to achieve privacy, Kim and Yang [16] introduce the concept of differential privacy to manipulate the features of images in a dataset with total randomness. Diluting unique features of each image makes the processed dataset immune to model inversion attacks [27] but reduces the utility of the entire dataset, resulting in a low inception score [28] on unsupervised CIFAR-10 [29].

2.2. Differential Privacy

Dwork et al. [17,30,31] first introduced differential privacy, an algorithm that captures the increased risk to one's privacy incurred by participating in a database. Nowadays, differential privacy is a reliable standard for privacy guarantees for algorithms on aggregate databases. Differential privacy for two neighboring datasets that differ by a single element is defined as follows:

A randomized mechanism, $M : D \to R$, with domain D and range R, satisfies (ϵ, δ)-differential privacy if for any two adjacent inputs $d, d' \in D$ and for any subset of outputs $S \subseteq R$, it holds that:

$$\Pr[M(d) \in S] \leq e^{\epsilon} \Pr[M(d') \in S] + \delta \tag{1}$$

The Gaussian and Laplace noise mechanisms [17] are commonly used to approximate a deterministic real-valued function, $f : D \to R$, via additive noise calibrated to f's sensitivity s_f, which is defined as the maximum of the absolute distance $|f(d) - f(d')|$, where d and d' are adjacent inputs. The Gaussian noise mechanism is defined as follows:

$$M(d) \triangleq f(d) + N(0, \sigma^2) \tag{2}$$

where $N(0, \sigma^2)$ is a normal (Gaussian) distribution with mean 0 and standard deviation σ. This mechanism satisfies (ϵ, δ)-differential privacy with $\sigma = \sqrt{2 \log(1.25/\delta)} \frac{s_f}{\epsilon}$. The Laplace noise mechanism is defined as follows:

$$M(d) \triangleq f(d) + Lap(0, b) \tag{3}$$

where $Lap(0, b)$ is a Laplace distribution with mean 0 and scale b. This mechanism satisfies $(\epsilon, 0)$-differential privacy with $b = \frac{s_f}{\epsilon}$.

2.3. Self-Attention

Vaswani et al. first introduced the self-attention mechanism as a particular case of their scaled-dot-product attention [18]. The input consists of queries and keys of dimension d_k and values of dimension d_v. They compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply the softmax function to obtain the weights on the values. For computational efficiency, they packed together the queries, keys, and values into matrices Q, K, and V. The matrix of outputs is defined as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

Compared to additive attention [32], scaled-dot-product attention is much faster and more space-efficient in practice because it can be implemented using a highly optimized matrix multiplication code. In a self-attention version of scaled-dot-product attention, the keys, values, and queries come from the same place, which, in the case of PPSGAN, is the final output of the encoder.

3. PPSGAN

Our PPSGAN methodology aims to protect the privacy of individuals related to an image dataset by generating a synthetic image dataset that can replace the original. The selective feature anonymization mechanism of PPSGAN effectively conceals the class-independent features and highlights the class-dependent features of each image.

3.1. Model Architecture

PPSGAN consists of a set of the encoder G_e, noise amplifier N_a, and decoder G_d and a set of the critic C_d and the classifier, C_c. The encoder G_e converts an input image x into its original feature vector z. The noise amplifier N_a adds noise to z using its novel anonymization mechanism. The decoder G_d reconstructs the modified feature vector $\tilde{z}$ to an anonymized image $\tilde{x}$. The critic C_d evaluates the quality of $\tilde{x}$, and the classifier C_c evaluates the utility of $\tilde{x}$.

The encoder G_e takes an image x of size $n \times n \times k$ and outputs a d-dimensional vector z. The decoder G_d reconstructs an image $\tilde{x}$ of size $n \times n \times k$ from the d-dimensional vector $\tilde{z}$. G_e is composed of four convolution layers with 5×5 kernel and stride 2, each followed by the batch normalization [33] and the LeakyReLU [34]. We also add a $4096 \times d$ dense layer after the last activation function to reduce the output to a d-dimensional vector z. G_d starts with a $d \times 4096$ dense layer that expands $\tilde{z}$ to fit the first deconvolution layer, followed by four sets of 5×5 kernel and stride 2 deconvolution (transposed convolution) layer [35], the batch normalization, and the LeakyReLU (the first three) or the sigmoid (the last).

The critic C_d and the classifier C_c take an image and share the four convolution layers with 5×5 and stride 2, each followed by the batch normalization and the LeakyReLU. In this method, C_d uses a 4096×1 dense layer with the sigmoid activation function for discrimination and C_c uses a 4096×10 dense layer with the softmax activation function for classification.

3.2. Noise Amplifier

The noise amplifier N_a adds noise to z. For N_a, we refine the original noise amplifier [16] using scaled-dot-product self-attention [18]. We first initialize the encoder G_e and the decoder G_d with a pretrained autoencoder to find the approximate sensitivity. In Kim and Yang's work [16], the approximate sensitivity s_e is defined as follows:

$$s_e = \max_{x_i, x_j \sim \mathbb{S}_t} |G_e(x_i) - G_e(x_j)| \tag{5}$$

where x_i and x_j are images sampled from the training set, $\mathbb{S}_t$, and $|\cdot|$ is the element-wise absolute value calculation of a vector. With s_e, the next step is to find the optimal scale vector σ^* for the initial noise vector α. Kim and Yang utilize the Gaussian and Laplace noise mechanisms [17] to find σ^*. With privacy budget hyperparameters ϵ and δ, σ^* is defined as follows:

$$\sigma* = \begin{cases} \sqrt{2\log(1.25/\delta)}\ \frac{s_e}{\epsilon} & \delta \neq 0 \\ \frac{s_e}{\epsilon} & \delta = 0 \end{cases} \tag{6}$$

If $\delta = 0$, we sample α from the Laplace distribution $Lap(0,1)$. Otherwise, we use the normal distribution $N(0,1)$. To find class-dependent features in z, we use scaled-dot-product self-attention [18]. The attention matrix is defined as follows:

$$Attention(z) = softmax(\frac{zz^T}{\sqrt{d}}) \tag{7}$$

We use $Attention(z)$ as a weight matrix to find class-dependent features. Note that our attention matrix $Attention(z)$ is the output of the softmax function. To find a weight matrix for class-independent

features, we subtract each value of $Attention(z)$ from 1 to reverse the weight. The negative-attention matrix is defined as follows:

$$NegAttention(z) = J - Attention(z) \tag{8}$$

where J is a matrix of ones that has the same dimension as the attention matrix. Now the final modified feature vector $\tilde{z}$ is defined as follows:

$$\tilde{z} = Attention(z) \cdot z + NegAttention(z) \cdot (\alpha \circ \sigma^*) \tag{9}$$

where $\cdot$ is the dot product and $\circ$ is the Hadamard product of two matrices. In our experiments, we use various combinations of ϵ and δ to train models with different levels of privacy.

3.3. Zero-Noise Penalty

The role of the decoder G_d is to rebuild a modified feature vector $\tilde{z}$ into its unique image form. Since the critic C_d and the classifier C_c evaluate the entire G_e - N_a - G_d network by the final output image, the performance of G_d also effects the training of the noise amplifier N_a. If G_d disregards the utility aspect, N_a amplifies the initial noise α as a whole to help G_d create realistic random images. The opposite case of G_d focusing too much on the utility can also happen. In this case, N_a cancels out α, feed G_d the raw feature vector z, and the G_e - N_a - G_d network works like an autoencoder.

To guide G_d in the right direction, we add a new term, zero-noise penalty L_{zero}, to the G_e - N_a - G_d network's loss function. The zero-noise penalty is the mean-squared-error loss between the original image and the reconstructed version of the original feature z using G_d as follows:

$$L_{zero} = [(x - G_d(z)]^2 \tag{10}$$

To calculate L_{zero}, we add a skip-connection [36] that jumps over N_a and directly connects G_e and G_d. Samples of $G_d(z)$ are in Figure 3.

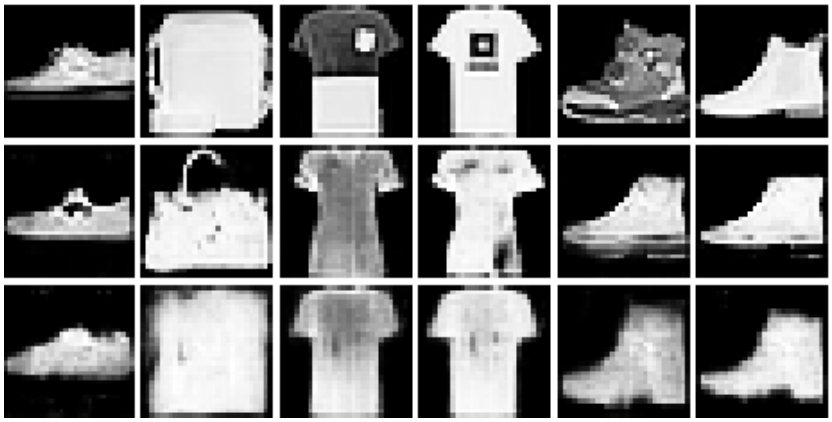


Figure 3. Samples from PPSGAN trained on the Fashion-MNIST dataset. Our model anonymizes each image (top) into a new image (middle). The zero-noise penalty is the mean-squared-error between the original (top) and the reconstructed version of the original feature z using the decoder G_d (bottom).

3.4. Adversarial Training

As depicted in Figure 1, the two sets of networks, G_e - N_a - G_d and C_d - C_c, are trained in an adversarial mode based on the training theme of ACGAN [9]. We first sample two batches of images x and x' from the real dataset $\mathbb{P}_r$. Then, the G_e - N_a - G_d network anonymizes x into $\tilde{x}$, and C_d - C_c compares $\tilde{x}$ and x'.

The critic C_d attempts to label $\tilde{x}$ as fake (0) and x' as real (1). The classifier C_c predicts class labels of x' and $\tilde{x}$. The objective function for C_d - C_c has two parts: the log-likelihood of correct discrimination, L_d, and the log-likelihood of correct class c, L_c.

$$L_d = E[\log P(D = 1 \mid x')] + E[\log P(D = 0 \mid \tilde{x})] \tag{11}$$

$$L_c = E[\log P(C = c \mid x')] + E[\log P(C = c \mid \tilde{x})] \tag{12}$$

Herein, C_d - C_c is trained to maximize $L_d + L_c$, while G_e - N_a - G_d is trained to maximize $L_c - L_d$.

In our implementation, we use the sigmoid-cross-entropy loss for L_d, the softmax-cross-entropy loss for L_c, and the zero-noise penalty to stabilize the training of G_d. Therefore, C_d - C_c learns to minimize $L_d + L_c$, and G_e - N_a - G_d learns to minimize $L_c - L_d + L_{zero}$.

After initializing G_e and G_d with a pretrained autoencoder and calculating the approximated sensitivity s_e, all the weights in PPSGAN are fine-tuned by carrying out an adversarial training in Algorithm 1. When the model converges, G_e - N_a - G_d is optimized to process any image sampled from the real dataset $\mathbb{P}_r$ to a synthetic image in the same class, with its novel selective feature anonymization.

Algorithm 1 PPSGAN training with default values of $n = 5$, $\eta = 1.0 \times 10^{-4}$, $\beta_1 = 0.5$, and $\beta_2 = 0.9$. $G(w_g, x)$ stands for G_e - N_a - G_d and $G_{zero}(w_g, x)$ stands for G_e - G_d.

Require: Initial G_e - N_a - G_d parameters w_{g0}, initial C_d - C_c parameters w_{c0}, and batch size m.

while w_g has not converged **do**
  for $t = 1, ..., n$ **do**
    for $i = 1, ..., m$ **do**
      Sample real data $x \sim \mathbb{P}_r$ and $x' \sim \mathbb{P}_r$.
      $\tilde{x} \leftarrow G(w_g, x)$
      $L_C^{(i)} \leftarrow L_d(w_c, \tilde{x}, x') + L_c(w_c, \tilde{x}, x')$
    $w_c \leftarrow AdamW(\nabla \frac{1}{m} \sum_{i=1}^{m} L_C^{(i)}, w_c, \alpha, \beta_1, \beta_2)$
  for $i = 1, ..., m$ **do**
    Sample real data $x \sim \mathbb{P}_r$ and $x' \sim \mathbb{P}_r$.
    $\tilde{x} \leftarrow G(w_g, x)$
    $\hat{x} \leftarrow G_{zero}(w_g, x)$
    $L_G^{(i)} \leftarrow L_c(w_c, \tilde{x}, x') - L_d(w_c, \tilde{x}, x') + [x - \hat{x}]^2$
  $w_g \leftarrow AdamW(\nabla \frac{1}{m} \sum_{i=1}^{m} L_G^{(i)}, w_g, \alpha, \beta_1, \beta_2)$

4. Experiments

We evaluate the performance of our model both quantitatively and qualitatively using the MNIST [37], Fashion-MNIST [38], CIFAR-10 [29], and SVHN [39] datasets. More details about each dataset are provided in Table 1. We first compare the classification accuracy of classifiers trained with the original, PPSGAN-processed, and ACGAN [9]-generated dataset. We also measure the sample diversity of the anonymized images using the Fréchet inception distance [40] on the CIFAR-10 dataset.

Table 1. Detailed information about datasets.

Dataset Name	Resolution	Training Set	Test Set
MNIST	$28 \times 28 \times 1$	60,000	10,000
Fashion-MNIST	$28 \times 28 \times 1$	60,000	10,000
CIFAR-10	$32 \times 32 \times 3$	50,000	10,000
SVHN	$32 \times 32 \times 3$	73,257	26,032

4.1. Experimental Details

Our implementation is done using TensorFlow [41]. We used a single NVIDIA Titan V GPU with a minibatch size of 256 and optimized our model with the AdamW optimizer [42] with a learning rate

of 1.0×10^{-4}, a weight decay of 1.0×10^{-4}, $\beta_1 = 0.5$, and $\beta_2 = 0.9$. ϵ and δ determine the privacy level of the noise amplifier. We use four combinations of (ϵ, δ): $(1.0, 0)$ and $(1.0, 1.0 \times 10^{-8})$ for standard privacy and $(0.1, 0)$ and $(0.1, 1.0 \times 10^{-8})$ for strong privacy. We indicate each model as PPSGAN-(ϵ, δ). The dimension size of the feature vectors z and $\tilde{z}$ is 128.

For ACGAN, we use the decoder G_d, the critic C_d, and the classifier C_c trained with the ACGAN training theme, feeding G_d with the noise sampled from $U[-1, 1]$ and class labels. We use the model structure of C_c for the utility evaluation classifier.

4.2. Utility Performance on Classifier Training

Publishing a useful synthetic image dataset necessitates a level of quality close to that of the original. In particular, we would like to know that if a classifier trained only with the PPSGAN-processed dataset still shows comparable performance to a classifier trained with the original. We first train PPSGAN and ACGAN using the training set in Table 1 and process or generate a synthetic dataset of the same size and class distribution. After training classifiers with the original or synthetic training sets, we measure the classification accuracy of each classifier with test sets. The performance results of the classifiers are listed in Table 2.

Table 2. Classification accuracies (%) of classifiers trained with the original or synthetic datasets. The PPSGAN-processed data show comparable performance to the original datasets, while the ACGAN-generated data show poor results.

Method	MNIST	Fashion-MNIST	CIFAR-10	SVHN
Original Data	99.60 ± 0.11	96.61 ± 0.02	88.80 ± 0.12	95.56 ± 0.03
ACGAN	98.15 ± 0.05	87.35 ± 0.34	59.58 ± 0.87	86.32 ± 0.67
PPSGAN-$(1.0, 0)$	98.23 ± 0.15	95.01 ± 0.08	76.91 ± 0.38	91.11 ± 0.01
PPSGAN-$(1.0, 1.0 \times 10^{-8})$	98.51 ± 0.20	95.07 ± 0.03	72.68 ± 0.04	90.85 ± 0.27
PPSGAN-$(0.1, 0)$	98.60 ± 0.11	94.53 ± 0.08	72.23 ± 0.03	90.56 ± 0.02
PPSGAN-$(0.1, 1.0 \times 10^{-8})$	98.24 ± 0.09	94.31 ± 0.21	72.36 ± 0.01	90.63 ± 0.01

This analysis shows that an anonymized dataset processed with PPSGAN preserves its original distribution and can replace the original dataset with a fair amount of utility. As shown in Table 2, our models with different privacy levels synthesize a dataset in sound quality for the classifier training, while synthetic datasets from ACGAN result in training a weak classifier. The selective feature anonymization is another strength of our model. The effect of a stronger privacy level is minimal because the majority of anonymization is applied to class-independent features.

4.3. Sample Diversity on CIFAR-10

We measure the inception score (IS) and the Fréchet inception distance (FID) of PPSGAN trained with CIFAR-10 to compare the sample diversity of PPSGAN with that of other published models. Lower inception scores and higher Fréchet inception distance indicate a lower sample diversity with a higher rate of mode collapse.

Kim and Yang [16] state that the randomness in the latent-space-level feature anonymization mechanism results in the low IS of PPAPNet, as shown in Table 3. The gap between our model and PPAPNet in the IS certifies that our selective feature anonymization is a more suitable method for privacy-preserving image data publishing. Our model also shows better results than ACGAN in the IS and FID as shown in Tables 3 and 4, respectively. PPSGAN-$(0.1, 0)$ shows 6.12 ± 0.05 of the IS, which is similar to that of DCGAN [11] and PPSGAN-$(0.1, 1.0 \times 10^{-8})$ shows 46.62 of the FID, similar to that of the residual flow method [43].

The IS and the FID of our model were obtained without significant optimization or fine-tuning of the hyperparameters for sample diversity. However, as analysed in the previous paragraph, our model produced either comparable or improved performance over other approaches. In fact,

our model produced fairly good results in the IS. We plan to improve PPSGAN for generating more diverse samples.

Table 3. Inception score on CIFAR-10.

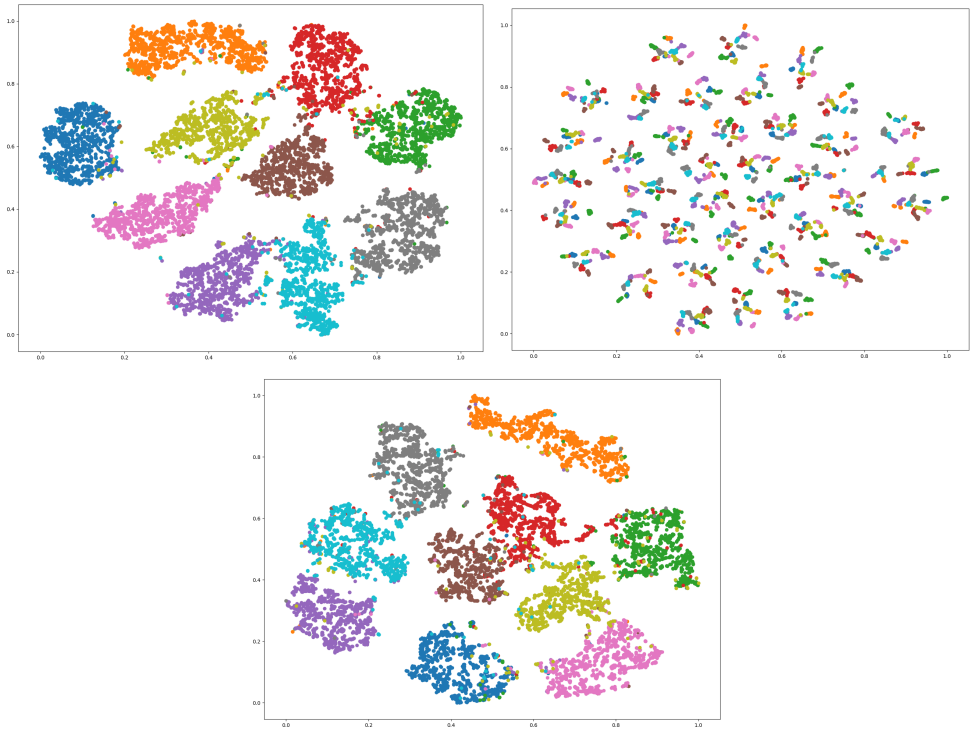
Method	Score
Original Data	11.31 ± 0.08
PPAP-s_e-(ϵ) [16]	2.83 ± 0.01
PPAP-s_e-(ϵ, δ) [16]	2.60 ± 0.01
ACGAN	5.70 ± 0.03
PPSGAN-$(1.0, 0)$	5.87 ± 0.13
PPSGAN-$(1.0, 1.0 \times 10^{-8})$	6.01 ± 0.08
PPSGAN-$(0.1, 0)$	6.12 ± 0.05
PPSGAN-$(0.1, 1.0 \times 10^{-8})$	5.91 ± 0.09
ALI [44] (in [45])	5.34 ± 0.05
BEGAN [8]	5.62
DCGAN [11] (in [46])	6.16 ± 0.07
Improved GAN (-L+HA) [28]	6.86 ± 0.06
WGAN-GP Resnet [19]	7.86 ± 0.07

Table 4. Fréchet inception distance on CIFAR-10.

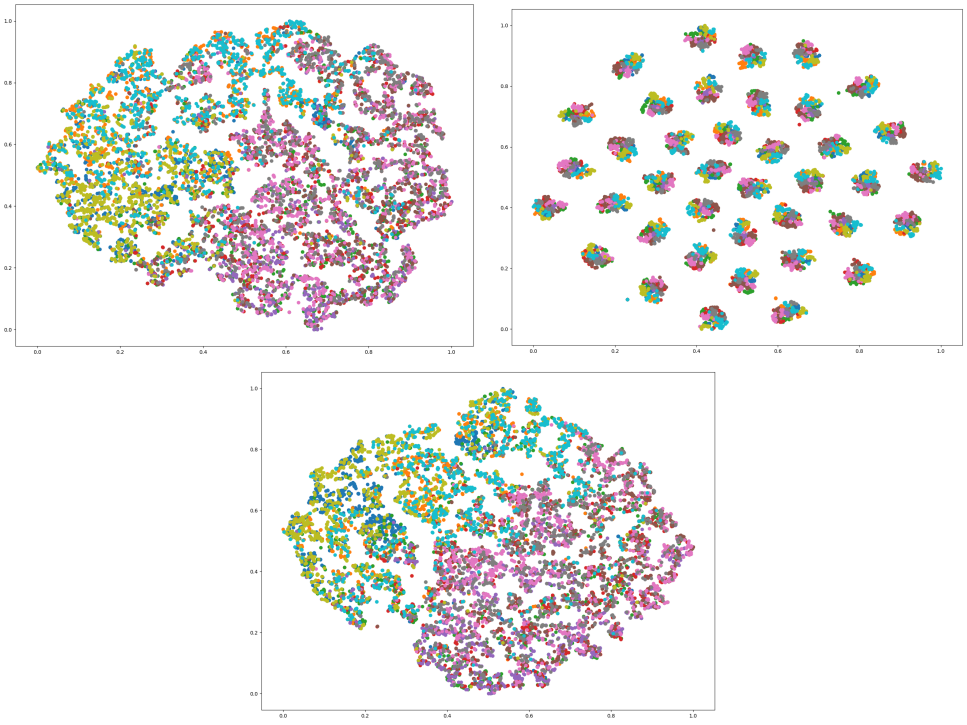
Method	Score
ACGAN	50.01
PPSGAN-$(1.0, 0)$	49.67
PPSGAN-$(1.0, 1.0 \times 10^{-8})$	47.67
PPSGAN-$(0.1, 0)$	48.99
PPSGAN-$(0.1, 1.0 \times 10^{-8})$	46.62
Residual Flow [43]	46.37
MSGAN [47]	28.73
FOGAN [48]	27.4
NCSN [49]	25.32
WGAN-GP + TT Update Rule [48]	24.8
SN-GANs [48]	21.7
MMD-GAN-rep [50]	16.21
AutoGAN [51]	12.42

4.4. t-SNE Visualization of the Latent Features

We present the t-SNE [52] visualizations of latent feature vectors of PPSGAN in Figure 4. The t-SNE visualizations of the original feature vector z, Figure 4 (left), show that the encoder G_e has learned to extract class-dependent features from the input image. In Figure 4 (right), the feature vectors of the anonymized image $G_e(\tilde{x})$ are also well clustered by the class-dependent features. Interestingly, the modified feature vector $\tilde{z}$, Figure 4 (middle), first form several clusters that do not correlate with class labels. Each cluster is then divided into smaller clusters according to each class label. As the noise amplifier N_a applies the normal or Laplace distribution-sampled noise to the class-independent features, class-dependent features also affect the t-SNE embedding. This unique cluster-in-cluster structure proves that N_a selectively adds noise to the class-independent features and preserves the class-dependent features.



(**a**) t-SNE visualizations of latent feature vectors on the MNIST.

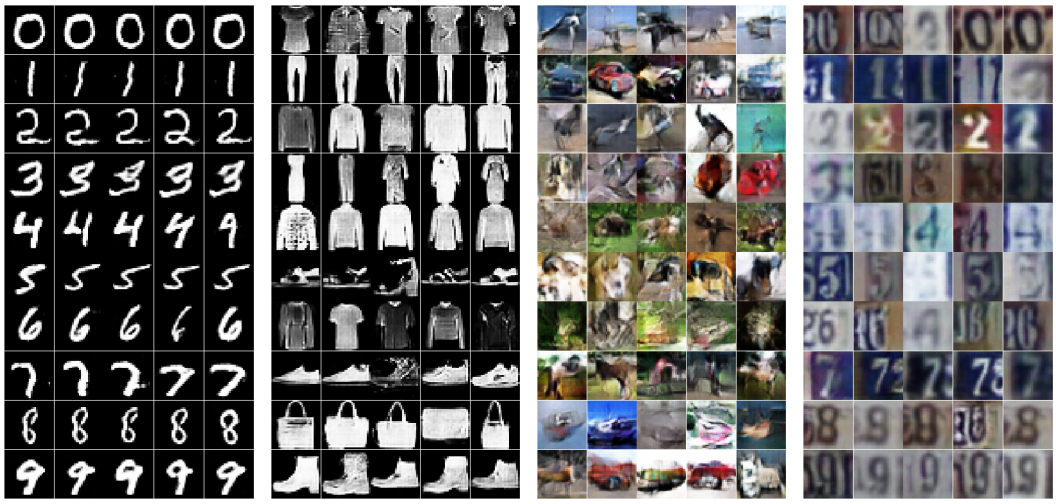


(**b**) t-SNE visualizations of latent feature vectors on the CIFAR-10.

Figure 4. t-SNE visualizations of latent feature vectors on the MNIST and CIFAR-10 datasets. Original feature vectors z (left), anonymized feature vectors $\tilde{z}$ (middle), and encoder-passed feature vectors of anonymized images $G_e(\tilde{x})$ (right). Both z and $G_e(\tilde{x})$ are well clustered according to each class label. The $\tilde{z}$ (middle) shows an interesting cluster-in-cluster structure, first clustered according to the class-independent features and then further clustered into smaller groups of each class label.

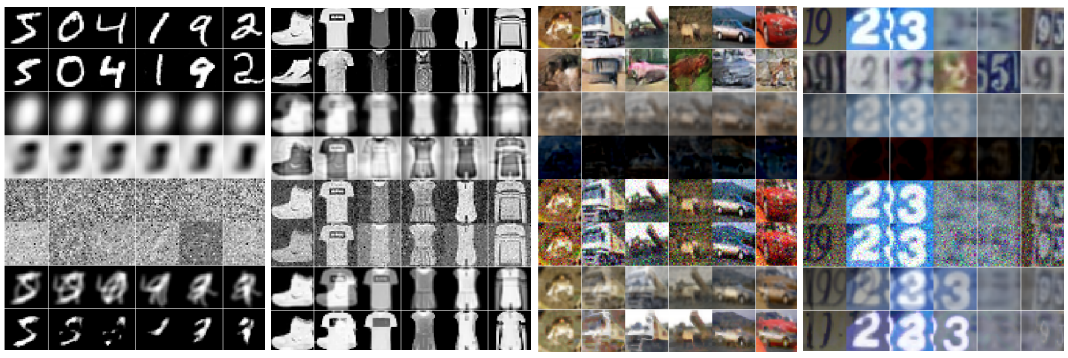
4.5. Anonymized Samples

We present five anonymized image samples of each class obtained with PPSGAN in Figure 5 and six representative samples in the MNIST, Fashion-MNIST, CIFAR-10, and SVHN datasets along with their anonymized versions modified with PPSGAN and other conventional image-processing methods in Figure 6. For conventional methods, we use Gaussian-blur, Laplace-blur, Gaussian-noise-adding, Laplace-noise-adding, uniform-filtering, and median-filtering. For each method, we train classifiers with various versions of training sets processed with different hyperparameters and choose the ones that show similar performance to the results of our models in Table 2. Compared to the realistic privacy-preserved images from our model, samples modified with conventional methods are either hard to recognize or still at privacy risk, as they maintain the unique features of the original. Samples of each class obtained with PPSGAN, as displayed in Figure 5, visually show that our PPSGAN methodology is a promising option for privacy-preserving image data publication.



(**a**) MNIST (**b**) Fashion-MNIST (**c**) CIFAR-10 (**d**) SVHN

Figure 5. Five samples of PPSGAN anonymized images for each class (row) in the MNIST, Fashion-MNIST, CIFAR-10, and SVHN datasets. These samples show that our PPSGAN anonymizes an image without losing the class-dependent features.



(**a**) MNIST (**b**) Fashion-MNIST (**c**) CIFAR-10 (**d**) SVHN

Figure 6. Six representative samples in the MNIST, Fashion-MNIST, CIFAR-10, and SVHN datasets, along with their anonymized versions modified with various processing methods. We list the original, PPSGAN-processed, Gaussian-blur, Laplace-blur, Gaussian-noise-adding, Laplace-noise-adding, uniform-filtering, and median-filtering (row). For conventional methods, we use hyperparameters that show comparable performance to PPSGAN in training classifiers. PPSGAN successfully preserves the class-dependent features and modifies the class-independent features. In contrast, samples modified with conventional methods are either hard to recognize or still at privacy risk, as they maintain the unique features of the original.

5. Conclusions

In this work, we present a privacy-preserving semi-generative adversarial network (PPSGAN), a methodology to selectively anonymize class-independent features of an image at the latent-space-level. In PPSGAN, a set of encoder–noise amplifier–decoder and a set of critic-classifier are trained in an adversarial mode to find the best way to modify an image in a privacy-preserving manner without losing its original class label. The noise amplifier plays a vital role in noise optimization and class-independent feature discrimination for adequate image anonymization. We evaluate the proposed PPSGAN with different metrics and datasets to demonstrate its potential.

In the future, we hope to strengthen our model with a deeper network structure to cover high-resolution image datasets, including ImageNet [53], CelebA [54], and LSUN [55]. We also intend to broaden the coverage of our novel selective feature anonymization methodology to a broader range of data domains, including video, text, and speech.

Author Contributions: Conceptualization, T.K.; methodology, T.K.; software, T.K.; validation, J.Y.; formal analysis, T.K.; investigation, T.K. and J.Y.; resources, J.Y.; data curation, T.K.; writing–original draft preparation, T.K.; writing–review and editing, J.Y.; supervision, J.Y.; project administration, J.Y.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No. 2018R1D1A1B07048790).

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

References

1. Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J.B.; Larochelle, H.; Zemel, R.S. Meta-Learning for Semi-Supervised Few-Shot Classification. *arXiv* **2018**, arXiv:1803.00676.
2. Santoro, A.; Bartunov, S.; Botvinick, M.M.; Wierstra, D.; Lillicrap, T.P. One-shot Learning with Memory-Augmented Neural Networks. *arXiv* **2016**, arXiv:1605.06065.
3. Socher, R.; Ganjoo, M.; Manning, C.D.; Ng, A. Zero-Shot Learning Through Cross-Modal Transfer. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 935–943.
4. Beaulieu-Jones, B.K.; Wu, Z.S.; Williams, C.; Lee, R.; Bhavnani, S.P.; Byrd, J.B.; Greene, C.S. Privacy-Preserving Generative Deep Neural Networks Support Clinical Data Sharing. *Circ. Cardiovasc. Qual. Outcomes* **2019**, *12*, e005122. [CrossRef] [PubMed]
5. Li, H.; Xiong, L.; Zhang, L.; Jiang, X. DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing. *Proc. VLDB Endow.* **2014**, *7*, 1677–1680. [CrossRef] [PubMed]
6. Zhang, J.; Cormode, G.; Procopiuc, C.M.; Srivastava, D.; Xiao, X. PrivBayes: Private Data Release via Bayesian Networks. *ACM Trans. Database Syst.* **2017**, *42*, 25:1–25:41. [CrossRef]
7. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Precup, D., Teh, Y.W., Eds.; PMLR: International Convention Centre: Sydney, Australia, 2017; Volume 70, pp. 214–223.
8. Berthelot, D.; Schumm, T.; Metz, L. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv* **2017**, arXiv:1703.10717.
9. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. *arXiv* **2016**, arXiv:1610.09585.
10. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661.
11. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
12. Zhao, J.J.; Mathieu, M.; LeCun, Y. Energy-based Generative Adversarial Network. *arXiv* **2016**, arXiv:1609.03126.
13. Chaudhuri, K.; Monteleoni, C.; Sarwate, A.D. Differentially Private Empirical Risk Minimization. *J. Mach. Learn. Res.* **2011**, *12*, 1069–1109. [PubMed]

14. Ma, L.; Sun, Q.; Georgoulis, S.; Gool, L.V.; Schiele, B.; Fritz, M. Disentangled Person Image Generation. *arXiv* **2017**, arXiv:1712.02621.
15. Ren, Z.; Lee, Y.J.; Ryoo, M.S. Learning to Anonymize Faces for Privacy Preserving Action Detection. *arXiv* **2018**, arXiv:1803.11556.
16. Kim, T.; Yang, J. Latent-Space-Level Image Anonymization With Adversarial Protector Networks. *IEEE Access* **2019**, *7*, 84992–84999. [CrossRef]
17. Dwork, C. Differential Privacy. *Automata, Languages and Programming*; Bugliesi, M., Preneel, B., Sassone, V., Wegener, I., Eds.; Springer: Berlin, Germany, 2006; pp. 1–12.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
19. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028.
20. Nowozin, S.; Cseke, B.; Tomioka, R. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In *Advances in Neural Information Processing Systems 29*; Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 271–279.
21. Kodali, N.; Abernethy, J.; Hays, J.; Kira, Z. On Convergence and Stability of GANs. *arXiv* **2017**, arXiv:1705.07215.
22. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. *arXiv* **2017**, arXiv:1703.05192.
23. Isola, P.; Zhu, J.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv* **2016**, arXiv:1611.07004.
24. Choi, Y.; Choi, M.; Kim, M.; Ha, J.; Kim, S.; Choo, J. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. *arXiv* **2017**, arXiv:1711.09020.
25. Zhu, J.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv* **2017**, arXiv:1703.10593.
26. Sun, Q.; Tewari, A.; Xu, W.; Fritz, M.; Theobalt, C.; Schiele, B. A Hybrid Model for Identity Obfuscation by Face Replacement. *arXiv* **2018**, arXiv:1804.04779.
27. Fredrikson, M.; Jha, S.; Ristenpart, T. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*; ACM: New York, NY, USA, 2015; pp. 1322–1333.
28. Salimans, T.; Goodfellow, I.J.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. *arXiv* **2016**, arXiv:1606.03498.
29. Krizhevsky, A.; Nair, V.; Hinton, G. CIFAR-10 (Canadian Institute for Advanced Research). Available online: http://www.cs.toronto.edu/~kriz/cifar.html (accessed on 3 May 2020).
30. Dwork, C. Differential Privacy: A Survey of Results. In *Theory and Applications of Models of Computation*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–19.
31. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*; Halevi, S., Rabin, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
32. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
33. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 448–456.
34. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv* **2015**, arXiv:1505.00853.
35. Noh, H.; Hong, S.; Han, B. Learning Deconvolution Network for Semantic Segmentation. *arXiv* **2015**, arXiv:1505.04366.
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
37. LeCun, Y.; Cortes, C. MNIST handwritten digit database 2010. Available online: yann.lecun.com/exdb/mnist (accessed on 3 May 2020).

38. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**, arXiv:1708.07747.
39. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Y Ng, A. Reading Digits in Natural Images with Unsupervised Feature Learning, *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. Available online: ufldl.stanford.edu/housenumbers (accessed on 3 May 2020).
40. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Proceedings of the Neural Information Processing Systems NIPS, Long Beach, CA, USA, 4–9 December 2017.
41. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Available online: tensorflow.org (accessed on 3 May 2020).
42. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101.
43. Chen, R.T.Q.; Behrmann, J.; Duvenaud, D.; Jacobsen, J.H. Residual Flows for Invertible Generative Modeling. *arXiv* **2019**, arXiv:1906.02735.
44. Dumoulin, V.; Belghazi, M.I.D.; Poole, B.; Lamb, A.; Arjovsky, M.; Mastropietro, O.; Courville, A. Adversarially Learned Inference. In Proceedings of the International Conference on Learning Representations ICLR, Toulon, France, 24–26 April 2017.
45. Warde-Farley, D.; Bengio, Y. Improving Generative Adversarial Networks With Denoising Feature Matching. In Proceedings of the International Conference on Learning Representations ICLR, Toulon, France, 24–26 April 2017.
46. Huang, X.; Li, Y.; Poursaeed, O.; Hopcroft, J.E.; Belongie, S.J. Stacked Generative Adversarial Networks. *arXiv* **2016**, arXiv:1612.04357.
47. Mao, Q.; Lee, H.Y.; Tseng, H.Y.; Ma, S.; Yang, M.H. Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. *arXiv* **2019**, arXiv:1903.05628.
48. Seward, C.; Unterthiner, T.; Bergmann, U.; Jetchev, N.; Hochreiter, S. First Order Generative Adversarial Networks. *arXiv* **2018**, arXiv:1802.04591.
49. Song, Y.; Ermon, S. Generative Modeling by Estimating Gradients of the Data Distribution. *arXiv* **2019**, arXiv:1907.05600.
50. Li, C.L.; Chang, W.C.; Cheng, Y.; Yang, Y.; Póczos, B. MMD GAN: Towards Deeper Understanding of Moment Matching Network, *arXiv* **2017**, arXiv:1705.08584.
51. Gong, X.; Chang, S.; Jiang, Y.; Wang, Z. AutoGAN: Neural Architecture Search for Generative Adversarial Networks. *arXiv* **2019**, arXiv:1908.03835.
52. van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008,** *9*, 2579–2605.
53. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the CVPR, Miami, FL, USA, 20–25 June 2009.
54. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015.
55. Yu, F.; Zhang, Y.; Song, S.; Seff, A.; Xiao, J. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv* **2015**, arXiv:1506.03365.