A Cycle-GAN Based Image Encoding Scheme for Privacy Enhanced Deep Neural Networks

David Rodriguez(✉) and Ram Krishnan(✉)

Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA
david.rodriguez3@my.utsa.edu, ram.krishnan@utsa.edu

Abstract. Deep learning model training on cloud platforms typically require users to upload raw input data. However, uploading raw image data to cloud service providers raises serious privacy concerns. To address this problem, we propose a Cycle-Gan based-image transformation scheme that leverages convolutional autoencoder image encoding for domain translation. Our Cycle-GAN based image transformation scheme enhances privacy of deep neural networks while preserving model utility. In this paper, we demonstrate that our Cycle-GAN based image transformation scheme protects visual feature information of sensitive image data. We evaluate the effectiveness of our proposed method to preserve model utility using classification accuracy and robustness against reconstruction attacks using structural similarity index measure (SSIM). The classification accuracy of encoded images using our proposed method is 92.48, 91.05, 90.37 for Chest X-ray, Dermoscopy and OCT datasets, respectively. The SSIM scores for reconstruction attacks where the attacker only has access to the encoded data and corresponding labels are 0.1002, 0.0995 and 0.0329 for Chest X-ray, Dermoscopy and OCT datasets, respectively. Our results demonstrate that the Cycle GAN based encoding scheme effectively enhance privacy while preserving model utility.

Keywords: Cycle-GAN · Deep Neural Networks · Convolutional Autoencoder · Privacy · Utility

1 Introduction

The amount of data generated by worldwide data sources has increased exponentially. Nevertheless, the utilization of big data is suboptimal without proper computing resources to extract patterns and vital information from zetabytes of data. Consequently, many businesses have switched to cloud service providers for computationally expensive tasks using large and complex datasets [1,2]. As a

Research supported in part by NSF CREST Grant HRD-1736209 (RK) and NSF CAREER Grant CNS-1553696 (RK).

V. Muthukkumarasamy et al. (Eds.): ICISS 2023, LNCS 14424, pp. 178–196, 2023.
https://doi.org/10.1007/978-3-031-49099-6_11

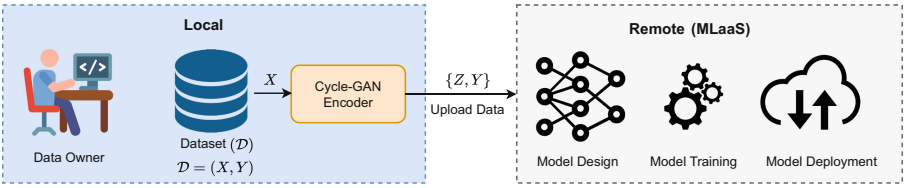

Fig. 1. Cycle-GAN based image transformation for privacy enhanced DNNs. Where X is the data owner's original dataset and $\{Z, Y\}$ are the data owner's encoded images and corresponding class labels. The encoded images and labels are uploaded to MLaaS provider for DNN model development and deployment while keeping the original image data private.

result, there has been a surge in the demand for cloud services. Cloud services are often categorized into Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) and IaaS offers infrastructure such as servers, virtual machines (VMs), storage, networks, operating systems on a pay-as-you-go basis. PaaS offers on-demand environments for developing, testing, delivering, and managing software applications. SaaS offers on-demand software applications over the internet which are typically on a subscription basis.

Additionally, machine learning-as-a-service (MLaaS) includes a variety of machine learning tools offered by cloud service providers such as Amazon, Google and Microsoft. MLaaS enables efficient model development and deployment at low cost. However, the adoption of MLaaS raises several data privacy concerns. This is especially true with sensitive image data e.g., suppose that a data owner uploads sensitive image data to an MLaaS provider for the purpose of developing a deep learning model but a curious MLaaS developer may also want to learn some additional sensitive information that could lead to identity theft, financial fraud, disease misdiagnosis [3–5]. Therefore, sensitive image data privacy plays an essential role in the deep learning life cycle.

The deep learning life cycle includes a training phase for model development and a testing phase for model deployment. Deep learning is susceptible to several attack methods during training and testing phase such as data poisoning attacks, model extraction attacks, model inversion attacks and adversarial attacks. However, in this work, we focus on protecting the privacy of sensitive image data for privacy enhanced deep neural networks (DNNs) during the training phase. Several image transformation methods have been proposed to protect the privacy of image data during the training phase of a deep learning life cycle [6–8]. However, a major challenge in transforming image data to enhance the privacy of DNNs is the trade-off between privacy and utility [9]. Typically, DNN model performance on original images degrades as images are transformed for privacy protection. To address this problem, we evaluate the effectiveness of Cycle-GAN [10] to preserve model utility using classification accuracy and robustness against reconstruction attacks using structural similarity index measure (SSIM).

In this paper, we propose a Cycle-GAN based image transformation scheme to enhance privacy of DNN model development and deployment on MLaas platforms as depicted in Fig. 1. Our Cycle-GAN method leverages autoencoder obfuscated images for domain translation. First, the autoencoder is trained to output visually unrecognizable versions of the original input image. Second, Cycle-GAN is trained to translate original images to the corresponding encoded image domain. We evaluate the robustness of our Cycle-GAN method to reconstruction attacks. In our results, we demonstrate that the proposed Cycle-GAN method enhances the privacy of image data while preserving model utility using Chest X-ray, Dermoscopy and OCT datasets.

In summary our contributions are as follows:

- We develop a Cycle-GAN based image transformation scheme for privacy enhanced deep neural networks.
- We enhance privacy of sensitive image data while maintaining classification accuracy.

The remainder of this paper is organized as follows. In Sect. 2, we provide an overview of related works for privacy enhancing methods in machine learning. In Sect. 3, we discuss the proposed Cycle-GAN method formulation and loss function. In Sect. 4, we describe the data sets, network architecture and training procedure. In Sect. 5, we evaluate our proposed Cycle-GAN method by analyzing the trade-off between privacy-utility and robustness to reconstruction attacks. Finally, we conclude our paper in Sects. 6.

2 Related Works

The security and privacy of machine learning models is usually concerned with the model's input, the model's output or the model itself. There are many proposed methods in the literature e.g., secure multi-party computation, homomorphic encryption, federated learning, visual image protection and learnable image encryption. Secure multi-party computation is a set of cryptographic protocols that allow multiple parties to evaluate a function to perform computation over each parties private data such that only the result of the computation is released among participants while all other information is kept private [11]. Secure multi-party computation methods have been applied in machine learning among multiple parties by computing model parameters using gradient descent optimization without revealing any information beyond the computed outcome [12–15]. The proposed Cycle-GAN image encoding scheme does not require multiple parties to compute the gradient descent of each model individually which is computationally expensive but instead enables users to encode private data individually and develop privacy enhanced deep neural networks with greater efficiency.

Homomorphic encryption is a type of encryption that allows multiple parties to perform computations on its encrypted data without having access to the original data [16–18]. It provides strong privacy but is computationally expensive requiring significant overhead to train machine learning models [19–21]. The proposed Cycle-GAN image encoding method does not use computationally expensive encryption operations or specialized primitives during model development.

Federated learning allows multiple parties to train a machine learning model without sharing data [22–24]. For example, in centralized federated learning a central server sends a model to multiple parties to train locally using their own data, then each participant sends it's own model update back to the central server to update the global model which is again sent to each party to obtain the optimal model without access to the local data by iterating through this process [25]. Essentially, federated learning builds protection into the model. Nevertheless, federated learning requires that each user have enough computing resources to train locally using their own data. The proposed Cycle-GAN image encoding method allows multiple parties to share obfuscated data for model training without the computational resource requirement of each participant.

Visual image protection methods transform plain images to unrecognizable encoded images while preserving important feature information for model utility. A few examples are pixelation, blurring, P3 [26], InstaHide [27] and NueraCrypt [28] which aim at preserving privacy and utility—a model trained on an encoded dataset should be approximately as accurate as a model trained on the original dataset [29,30]. InstaHide mixes multiple images together with a linear pixel blend and randomly flips the pixel signs. NeuraCrypt encodes data instances through a neural network with random weights and adds position embeddings to keep track of image structure then shuffles the modified output in blocks of pixels. However, [31] showed that position information, permutation order and image-encoding pairs could be learned given an unordered set of images and corresponding encodings. The proposed Cycle-GAN image encoding method inherently generates encoded images by learning a mapping function between original images and distorted images while reducing data leakage during domain translation.

Learnable image transformation methods obfuscate image data such that the encoded versions are useful for classification [6–8,32,33]. However, in some cases network adjustments are required to process learnable image transformations such as blockwise adaptation [6]. Our proposed Cycle-GAN encoding scheme does not require any particular changes to the network to develop models using the encoded data. Our work is most closely related to [34] but instead of transforming image data using adversarial perturbations for domain translation we develop our encoding model leveraging obfuscated autoencoder output. The key benefit in our method is that the autoencoder is specifically optimized to generate transformed images that retain image features that are useful for model utility.

3 Cycle-GAN Image Transformation Formulation

We aim to transform image data using a Cycle-GAN based approach to obfuscate sensitive feature information while preserving classification accuracy. The proposed method allows participants within a network to share sensitive image data while protecting privacy and maintaining model utility. We consider features that do not highly contribute to the classification task as sensitive features.

For example, in chest x-ray images the features that do not highly contribute to the classification of the pneumenia disease are considered sensitive features. On the other hand, we consider features that highly contribute to the classification task as non-sensitive features. For example, in chest x-ray images the features that highly contribute to the classification of the pneumenia disease are considered non-sensitive features. Our goal is to transform image data such that non-sensitive features are preserved for image classification. We aim to preserve classification accuracy of transformed images similar to original images.

Our goal is to enhance the privacy of deep neural networks by transforming image data using Cycle-GAN for image-encoding domain translation. Let $\mathcal{X}$ be the set of all images in the data domain, $X \subseteq \mathcal{X}$ is the local subset of private images and Y is the corresponding label set. Given the private image dataset $\{x_i\}_{i=1}^{N}$ where $x_i \in X$, the images are transformed using the private Cycle-GAN encoding function $G_Z(x)$. The encoded images and corresponding labels can be safely uploaded to remote MLaaS providers for deep learning model development using visibly unrecognizably images. The proposed Cycle-GAN method is similar to [34] but instead of transforming image data using adversarial perturbations for domain translation we develop our encoding model leveraging modified autoencoder output. The proposed method consists of a classification model to distinguish between non-sensitive features. Additionally, our methods consists of an autoencoder model for initial image transformation. Finally, the encoding network consists of a Cycle-GAN model for final image transformation. The training objective is to optimize the model parameters of generator G_Z to transform original images into encoded images.

3.1 Overview

First, the classification model is trained to classify original images using a constructed dataset that follows the probability distribution of the original dataset and their respective class labels. Our objective function for the classification model has a loss term for classifying non-sensitive features. The goal is to classify non-sensitive features of a given image with high classification accuracy. Second, the classification model loss function is used to optimize the model parameters of a randomly initialized autoencoder network given it's output to generate distorted versions of the input image while preserving important feature information for model utility. Third, the Cycle-GAN network is used to transform original images to the distorted images. Our Cycle-GAN based image transformation final objective function follows original Cycle-GAN [10] objective which contains three loss terms: adversarial loss for mapping original images to encoded images, adversarial loss for mapping encoded images to original images and cycle consistency loss to prevent the learned mappings from contradicting each other. We aim to learn a mapping function from original images to distorted images to transform private data while preserving important feature information for model utility.

Our proposed Cycle-GAN image transformation scheme consists of a non-sensitive feature loss, distortion loss, adversarial loss and cycle consistency loss.

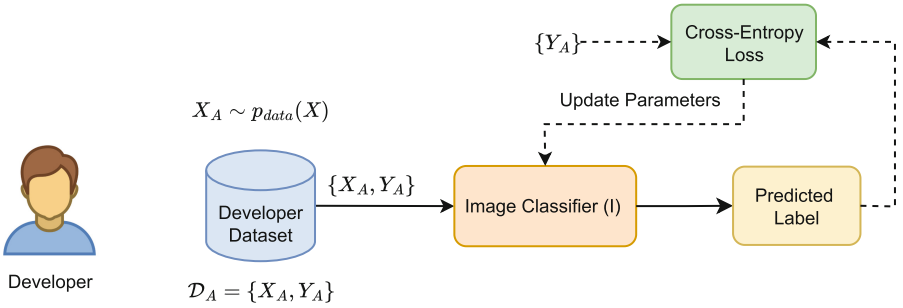

Fig. 2. Image classification model training phase. Where X_A is the developer's constructed data set that follows the probability distribution of the original dataset and Y_A are the corresponding class labels. Standard DNN image classification model training is conducted to predict the labels of non-sensitive image features.

First, we develop a classification model to classify non-sensitive features using a non-sensitive feature loss as depicted in Fig. 2. Second, we develop an autoencoder model to distort images using an image distortion loss as depicted in Fig. 3. The networks are trained using a constructed dataset that follows the probability distribution of the original dataset i.e., $x_a \sim p_{data}(x)$. The non-sensitive feature loss is used to minimize the error between the true label and the classifier's predicted label for non-sensitive features. For example, the true label of a chest x-ray image is the correct class assigned to the image which specifies whether the image has pneumonia disease or not. The distortion loss is used to minimize the error between the true label and the pre-trained classifier's predicted output label given the autoencoder distorted image for each sample in the constructed dataset. The aim is to distort image data and classify non-sensitive features with high classification accuracy. Third, we train Cycle-GAN to using adversarial loss and cycle consistency loss to learn a mapping function from images to distorted images as depicted in Fig. 4.

3.2 Non-sensitive Feature Loss

The non-sensitive feature loss function L_n uses cross-entropy to measure the performance of the image classifier I which is trained to classify non-sensitive features.

$$L_n(I, X_A, Y_A) = -\frac{1}{N}\sum_{i=1}^{N} Y_{A_i} log(I(X_{A_i})) \tag{1}$$

where X_{A_i} is the i^{th} image and Y_{A_i} is the corresponding ground truth identity label. $I(X_{A_i})$ is the image classifier's predicted output for the i^{th} image.

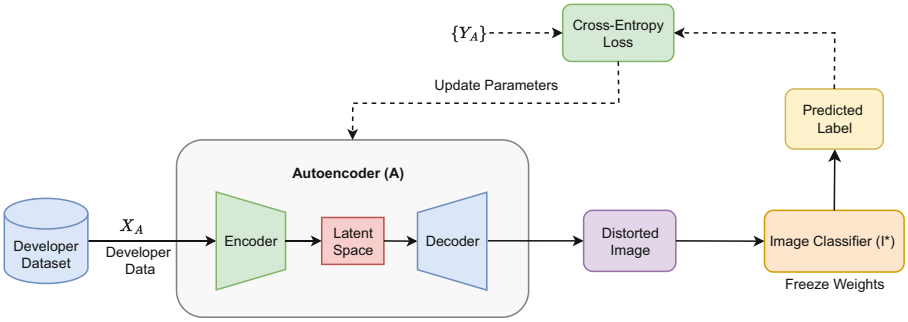

Fig. 3. Distortion model training phase. An autoencoder network A is optimized to generate distorted images that preserve important feature information for model utility given the developer input data. The pre-trained classifier I^* model parameters are frozen to ensure they remain constant during the distortion model training. The error between the predicted output label of I^* given distorted images and the true label is minimized. The model parameters of the A are updated based on the gradient of the crossentropy loss.

3.3 Non-sensitive Feature Loss Objective

The goal is to find the classification model I parameters that minimize the error between the true label and predicted label.

We aim to solve:

$$I^* = \underset{I}{\operatorname{argmin}} L_n(I, X_A, Y_A) \tag{2}$$

3.4 Distortion Loss

The distortion loss function L_d uses cross-entropy to distort feature information in sensitive image data.

$$L_d(A, I^*, X_A, Y_A) = -\frac{1}{N}\sum_{i=1}^{N} Y_{A_i} log(I^*(A(X_{A_i}))) \tag{3}$$

where A is a randomly initialized autoencoder network and I^* is a pre-trained image classification function. $I(A(X_{A_i}))$ is the image classifier's predicted output given the i^{th} distorted image.

3.5 Distortion Loss Objective

The goal is to find the autoencoder model A parameters that minimize the error between the true label and the image classifier I predicted output given the i^{th} distorted image i.e., $I^*(A(X_{A_i}))$.

We aim to solve:

$$A^* = \underset{A}{\operatorname{argmin}} L_d(A, I^*, X_A, Y_A) \tag{4}$$

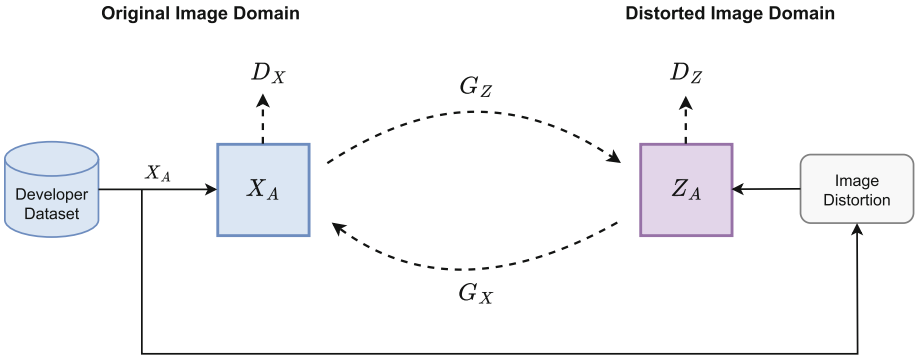

Fig. 4. Cycle-GAN based image transformation for privacy enhanced DNNs. Where X_A is the developer data set and Z_A is the corresponding distorted image generated using the pre-trained distortion model A^*. Generator G_Z learns a mapping function from X_A to Z_A and generator G_X learns a mapping function from Z_A to X_A. Discriminator D_X distinguishes between real and fake images while discriminator D_Z distinguishes between real and fake distorted images.

3.6 Adversarial Loss

The full adversarial loss consists of a loss term from generators (G_Z, G_X) and discriminators (D_Z, D_X). The following equations describe the adversarial loss term.

$$\begin{aligned} L_{GAN}(G_Z, D_Z, X_A, Z_A) = \mathbb{E}_{z_a \sim p_{enc}(z_a)}[\log D_Z(z_a)] \\ + \mathbb{E}_{x_a \sim p_{data}(x_a)}[\log(1 - D_Z(G_Z(x_a)))] \end{aligned} \tag{5}$$

where G_Z tries to generate encoded images $G_Z(x_a)$ that are similar to distorted autoencoder images i.e., $z_a = A^*(x_a)$. D_Z distinguishes between real autoencoder distorted images z_a and generated encoded images $G_Z(x_a)$. G_Z minimizes the objective while D_Z maximizes the objective, $\min_{G_Z} \max_{D_Z} L_{GAN}(G_Z, D_Z, X_A, Z_A)$.

$$\begin{aligned} L_{GAN}(G_X, D_X, Z_A, X_A) = \mathbb{E}_{x_a \sim p_{data}(x_a)}[\log D_X(x_a)] \\ + \mathbb{E}_{z_a \sim p_{enc}(z_a)}[\log(1 - D_X(G_X(z_a)))] \end{aligned} \tag{6}$$

where G_X tries to images $G_X(z_a)$ that are similar to original images x_a, while D_X distinguishes between the real original images and generated images $G_X(z_a)$. G_X minimizes the objective while D_X maximizes the objective, $\min_{G_X} \max_{D_X} L_{GAN}(G_X, D_X, Z_A, X_A)$.

3.7 Cycle Consistency Loss

The cycle consistency loss term is computed using generator G_Z and generator G_X. First, the original images X are translated into the distorted image domain Z_A using generator G_Z. Then the generated distorted image is translated back

into the original image domain using generator G_X i.e., forward cycle. Second, the autoencoder distorted image Z_A is translated into the original image X domain using generator G_X. Then the generated original image is translated back into distorted image domain using generator G_Z i.e., backward cycle. The mean absolute error between the original images and the forward cycled images is computed. The mean absolute error between the distorted images and the backward cycled images is computed.

The computed cycle consistency loss values for the original images and distorted images are summed together below.

$$\begin{aligned} L_{cyc}(G_Z, G_X) = \mathbb{E}_{x_a \sim p_{data}(x_a)}[||G_X(G_Z(x_a)) - x_a||_1] \\ + \mathbb{E}_{z_a \sim p_{enc}(z_a)}[||G_Z(G_X(z_a)) - z_a||_1] \end{aligned} \tag{7}$$

$G_X(G_Z(x_a))$ is the forward cycled original image and $G_Z(G_X(z_a))$ is the backward cycled distorted image. The error between the cycled images and real images is minimized and summed to compute the total cycle consistency loss.

3.8 Cycle-GAN Encoding Full Objective

The adversarial and cycle consistency loss terms are summed together for the full objective. The full objective for the Cycle-GAN encoding loss consists of an adversarial loss term and cycle consistency loss term.

The full objective is:

$$\begin{aligned} L(G_Z, G_X, D_X, D_Z) = L_{GAN}(G_Z, D_Z, X_A, Z_A) \\ + L_{GAN}(G_X, D_X, Z_A, X_A) \\ + \lambda L_{cyc}(G_Z, G_X) \end{aligned} \tag{8}$$

where λ controls the importance of the objectives. We solve the following optimization problem:

$$G_Z^*, G_X^* = \underset{G_Z, G_X}{\operatorname{argmin}} \max_{D_Z, D_X} L(G_Z, G_X, D_X, D_Z) \tag{9}$$

4 Methods

4.1 Dataset

In this work, we use three publicly available medical image datasets to develop our Cycle-GAN encoding scheme, which include Chest X-Ray, Dermoscopy and Optical Coherence Tomography (OCT). The Chest X-ray dataset [35] consists of 5,863 grayscale chest radiograph images used to diagnose thorax disease. It includes two classes, where each image is labeled as "Pneumonia" or "Normal". The Dermoscopy dataset [36] contains 17.8K color images of skin lesions, which are used to diagnose melanoma skin cancer. It includes two classes, where each image is labeled as "Melanoma" or "NotMelanoma". We consider all non-melanoma images to be part of the NotMelanoma class [37]. The OCT dataset

[35] consists of 84,495 grayscale images with four classes—including "Choroidal Neovascularization (CNV)", "Drusen", "Diabetic macular edema (DME)", and "Normal". It utilizes light waves to take cross-section imagery of the retina to assist in diagnosing retina disease and disorders in the optic nerve.

4.2 Network Architecture

The Cycle-GAN based image encoding architecture consists of three parts: a Resnet-50 for image classification, a standard convolutional autoencoder (CAE) for image distortion and a Cycle-GAN for image encoding. Resnets are large state-of-the-art DL architectures that consist of several blocks of residual modules and skip connections [38]. The classification model architecture consists of a Resnet-50 network trained to classify non-sensitive features i.e., features that highly contribute to the classification task. The CAE network consists of three convolution layers with 32, 64 and 128 filters, respectively. The kernel size is 3×3 with a stride of 2 and a latent space of 128. Each convolution layer consists of a leaky relu activation function with alpha 0.2 followed by a batch normalization layer. The decoder network consists of three transposed convolution layers with 128, 64 and 32 filters, respectively. The kernel size is 3×3 with a stride of 2 and output size of $224 \times 224 \times 3$. Each transposed convolution layer consists of a leaky relu activation function with alpha 0.2 followed by a batch normalization layer.

The Cycle-GAN network consists of two generators (G_Z, G_X), two discriminators (D_Z, D_X). The generator networks contain three convolutions, 9 residual blocks [38], two transpose convolutions and one convolution that maps features to RGB. Also, we use instance normalization [39]. Similar to [10] we use 70×70 PatchGANs for the discriminator networks. Generator G_Z is used to translate original images to the distorted image domain and generator G_X is used to translate distorted images to original image domain. Discriminator D_Z is used to distinguish between real and fake distorted images and discriminator D_X is used to distinguish between the real and fake original images.

4.3 Training Procedure

Image Classification Model. Our training procedure consists of an image classification phase to classify non-sensitive features, an image distortion phase to obfuscate sensitive image data and an image encoding phase to reduce the risk of data leakage. First, in the image classification phase we train a Resnet-50 model from randomly initialized parameters using the original image dataset and corresponding class labels for non-sensitive features. We train using binary crossentropy loss function for dataset with two classes. Additionally, we train using categorical crossentropy loss function for dataset with more than two classes. We wish to classify non-sensitive features for a given data set i.e., features that are strongly correlated with the class label. The classification model loss function is used to optimize our image distortion model.

Image Distortion Model. Second, in the image distortion phase we randomly initialize the CAE model parameters and add its output to the pre-trained Resnet-50 classification model input for each of the given original images. We freeze the Resnet-50 classifier model parameters to ensure that the weights do not change during training for the image distortion phase. During training we use the classification model loss function to find the CAE model parameters that minimize the error between the true label and the image classifier predicted label given the distorted image for the original image dataset. We wish to preserve non-sensitive feature information while reconstructing an unrecognizable version of the original image. The reconstructed image is a distorted version of the original image that is useful for classification. It is generated to obfuscate sensitive image data. To obfuscate sensitive image data we use the output of the image classifier to optimize the CAE model with crossentopy loss function.

Cycle-GAN Encoding Model. Third, in the Cycle-GAN encoding phase we learn a mapping function between original images and distorted images. The distorted images are generated using the pre-trained autoencoder i.e., $Z_A = A^*(X_A)$. The Cycle-GAN adversarial loss is computed using generator G_Z, generator G_X, discriminator D_Z and discriminator D_X. Generator G_Z is used as a mapping function from the original image domain X_A to the distorted image domain Z_A. Discriminator D_Z is a binary classifier used to distinguish between real distorted images Z_A and generated distorted images $G_Z(X_A)$. Generator G_Z wishes to minimize the probability of $G_Z(X_A)$ being classified as generated distorted images by discriminator D_Z while D_Z aims to maximize the probability of the real distorted images Z_A being classified as real and generated distorted images $G_Z(X_A)$ being classified as fake. The aim is to learn a generator G_Z that translates original images X_A into the distorted image domain.

Discriminator D_X is a binary classifier used to distinguish between real and generated original images. We obtain generated original images using generator G_X given distorted images as input to generator G_X, i.e. $G_X(Z_A)$. Generator G_X wishes to minimize the probability of $G_X(Z_A)$ being classified as a generated original image by discriminator D_X while D_X aims to maximize the probability of the real original images X_A being classified as a real and generated original images $G_X(Z_A)$ being classified as fake. As a result, we learn a generator that translates original images into the distorted image domain.

Generator G_Z and generator G_X are used to compute the cycle consistency loss. The original images are translated into the distorted image domain and then back to the original image domain which is called a forward cycle i.e., $G_X(G_Z(X_A))$. Then the distorted images are translated into the original image domain and then back to the distorted image domain which is called a backward cycle i.e., $G_Z(G_X(Z_A))$. The mean absolute error between the original images and the forward cycled images is computed. The mean absolute error between the distorted images and the backward cycled images is computed. Both values are summed to ensure that the real and generated images remain similar.

All networks were trained using the adam optimizer with a batch size of 32. We utilize check points to save the model with the highest validation accuracy during model development. The classification and autoencoder models were trained for 100 epochs and Cycle-GAN network was trained for 200 epochs. During Cycle-GAN training we set $\lambda = 10$. All images were resized to 224×224 and normalized between 0 and 1. Each dataset was randomly shuffled and split ten times to generate multiple subsets of the train, test and validation set. Each network was trained ten times for a given dataset to assess the average performance of all models across multiple subsets of the data.

5 Evaluation

5.1 Evaluating Privacy/Utility Trade-Off

We develop classification models using Resnet-50 architecture as described in Sect. 4.2 and encoded images generated by the proposed Cycle-GAN encoding scheme. Additionally, we also develop classification models using Resnet-50 architecture and original images to evaluate the trade-off between privacy and model utility, i.e. we measure the change in classification accuracy for a network trained with original images compared to a network trained with encoded images. First, we transform the original images using our Cycle-GAN image transformation method. Second, we compare the classification accuracy of original images and the transformed images. The classification accuracy of networks trained using Cycle-GAN transformed images exhibits a slight performance decrease compared to networks trained using original images. To quantify the trade-off between privacy and utility we measure the reduction in classification accuracy for the network trained using original images and the network trained using encoded images. Additionally, we measure the SSIM score between original images and encoded images. SSIM measures similarities within pixels i.e., it checks whether the pixels in the images line up and or if the images have similar pixel density values. In our experiments, we demonstrate that the proposed Cycle-GAN method allows us to maintain high classification accuracy of $92.48 \pm 1.53\%, 91.05 \pm 1.10\%, 90.37 \pm 2.06\%$ for Chest X-ray, Dermoscopy and OCT datasets, respectively, compared to models trained using plain images with classification accuracy of $96.90 \pm 1.26\%, 95.20 \pm 0.85\%, 95.20 \pm 2.71\%$ for Chest X-ray, Dermoscopy and OCT datasets, respectively which is similar to original images as shown in Table 1. Additionally, we demonstrate that the proposed Cycle-GAN method enhances privacy using SSIM scores between original and encoded images. The SSIM scores closer to zero indicate that the images are highly dissimilar. The SSIM scores in our privacy versus utility experiments were 0.0935, 0.0582, 0.0277 for Chest X-ray, Dermoscopy and OCT datasets, respectively,

Table 1. Trade-off between privacy and model utility for medical image deep learning models. Classification accuracy slightly decreases for networks trained using encoded medical images compared to networks trained using plain images i.e., original images. The proposed scheme enhances privacy of medical image DNNs while preserving model utility.

Encoding Scheme	Classification Acc.%		
	Chest X-ray	Dermoscopy	OCT
Plain Images	96.90 ± 1.26	95.20 ± 0.85	95.20 ± 2.71
Proposed Method	92.48 ± 1.53	91.05 ± 1.10	90.37 ± 2.06

5.2 Evaluating Robustness to Attacks

Model Stealing Attack. We evaluate the robustness of our proposed Cycle-GAN based image encoding method against reconstruction attacks given the assumption that the data owner's Cycle-GAN encoder is publicly available to an attacker. The goal of an attacker is to learn G_X given G_Z. In this case, the attacker begins training Cycle-GAN by querying G_Z using his own constructed dataset X_B to obtain the predicted output. $G_Z(X_B)$ is used to train the attacker's generator F_X which is a randomly initialized version of G_X. Additionally, the attacker randomly initializes two discriminators Q_X to distinguish between real and fake images and Q_Z to distinguish between real and fake distortions. Following the previously discussed Cycle-GAN standard training procedure, the attacker learns a mapping function F_X^* to reconstruct the data owner's original image dataset. During training, we freeze the weights of the data owner's original encoder G_Z.

Model Stealing Attack Results. We evaluate the performance of the model stealing attack using structural similarity index measure (SSIM). The SSIM values that are closer to 1 indicate that the reconstructed images are similar to the original images and values closer to 0 indicate that reconstructed images are poor quality compared to original images. The model stealing attack SSIM scores are shown in row 1 of Table 2. The model stealing attack SSIM scores for Chest X-ray, Dermoscopy and OCT datasets are 0.6064, 0.7783 and 0.5981, respectively. It is evident from our SSIM results that the attacker can reconstruct the data owner's original dataset with poor quality given that he has access to data owner's original encoder G_Z. The model stealing attacks is a baseline attack method with the strong assumption that an attacker has access to the data owner's original encoding function.

Minimal Data Subset Attack. We evaluate the robustness of our proposed Cycle-GAN image encoding method against minimal data subset attacks where the adversary is granted access to a subset of the data owner's original image

Table 2. Proposed Cycle-GAN image reconstruction attack SSIM results. SSIM scores near 1 indicate high quality image reconstruction whereas scores closer to 0 indicate poor quality imge reconstruction.

Attack Method	Attacker's Knowledge	SSIM Score		
		Chest X-ray	Dermoscopy	OCT
Model Stealing	G_Z, Z, Y	0.6064	0.7783	0.5981
Min. Data Subset (FT)	X, Z, Y	0.7582	0.8154	0.7109
Min. Data Subset (RI)	X, Z, Y	0.7461	0.8033	0.7082
Cycle GAN Recon	Z, Y	0.1002	0.0995	0.0329

dataset and corresponding encoded images. The goal is to develop a deep learning model to reconstruct original images given encoded images. The attack is performed by incrementally updating the model parameters using a single image-encoding pair from the data owner's original dataset and corresponding encoded images i.e., $\{X, Z\}$. Image-encoding pairs are gradually included during the training process until SSIM saturates. The image reconstruction model parameters are updated by minimizing the mean squared error between original images and reconstructed images given the encoded samples. At the conclusion of each training step, we measure the SSIM score of the data owner's original images and the reconstructed images. First, the attacker develops a randomly initialized (RI) reconstruction model with a subset of the data owner's original image-encoding pair. Second, the attacker pre-trains a reconstruction model using his constructed dataset X_B and later fine-tunes the network (FT) with a subset of the data owner's original image-encoding pair. Afterwards, the reconstruction model is used to reconstruct the data owner's original image dataset.

Minimal Data Subset Attack Results. The reconstruction model performance is evaluated using SSIM. The SSIM results reflect the model performance as SSIM scores begins to saturate. The fine-tuned reconstruction model SSIM scores are shown in row 2 of Table 2. The fine-tuned SSIM scores for Chest X-ray, Dermoscopy and OCT datasets are 0.7582, 0.8154 and 0.7109, respectively. The randomly initialized reconstruction model SSIM scores are shown in row 3 of Table 2. The fine-tuned SSIM scores for Chest X-ray, Dermoscopy and OCT datasets are 0.7461, 0.8033 and 0.7082, respectively. The SSIM scores are indicative of good quality image reconstruction. The minimal data subset attack is a baseline attack method in which the attacker has access to a subset of the original image-encoding pairs.

Reconstruction Cycle GAN Attack. We evaluate the robustness of our proposed method against Cycle-GAN reconstruction attacks. In this case, an attacker constructs a dataset that follows the probability distribution of the

data owner's original dataset i.e., $x_b \sim p_{data}(x)$ and attempts to reconstruct the original dataset by learning his own mapping function using a Cycle-Gan based approach. First, the attacker develops his own image classification model using the constructed dataset and corresponding class labels by following the previously mentioned procedure from our proposed method. Second, the attacker develops a distortion model using the constructed dataset and corresponding class labels by following the previously mentioned procedure from our proposed method. Third, the attacker develops a Cycle-GAN encoding model using the constructed dataset and corresponding class labels by following the previously mentioned procedure from our proposed method. The goal is to learn a mapping function between the attacker's constructed dataset and the attacker's distorted dataset. We assume an attacker only has access to the data owner's encoded dataset and corresponding labels.

The attacker's reconstruction Cycle-GAN attack network consists of the same components of the proposed method i.e., two generators (F_Z, F_X), two discriminators (Q_Z, Q_X). The attacker's distorted dataset Z_B is generated using pre-trained distortion model $Z_B = A_B^*(X_B)$. Generator F_Z is used to translate the attacker's constructed images to the distorted image domain and generator F_X is used to translate distorted images to the attacker's constructed image domain. Discriminator Q_Z is used to distinguish between the real and fake distorted images and discriminator Q_X is used to distinguish between the attacker's real and fake constructed images.

Reconstruction Cycle-GAN Adversarial Loss. The adversarial loss term is computed using generator F_Z, generator F_X, discriminator Q_Z and discriminator Q_X. Discriminator Q_Z is a binary classifier used to distinguish between the distorted set Z_B and the generated distorted set $F_Z(X_B)$. First, generator F_Z is used as a mapping function from the attacker's constructed image domain to the distorted image domain $Z'_B = F_Z(X_B)$. Generator F_Z wishes to minimize the probability of Z'_B being classified as a generated distorted image by discriminator Q_Z while Q_Z aims to maximize the probability of the real distorted images Z_B being classified as real and generated distorted images Z'_B being classified as fake. The attacker learns a generator F_Z that translates constructed images X_B into the distorted image domain.

Discriminator Q_X is a binary classifier used to distinguish between real and generated constructed images. We obtain X'_B using generator F_X given the distorted set as input to generator F_X, i.e. $X'_B = F_X(Z_B)$. Generator F_X wishes to minimize the probability of X'_B being classified as a generated construct image by discriminator Q_X while Q_X aims to maximize the probability of real constructed images X_B being classified as a real and generated constructed images X'_B being classified as fake. The attacker learns a generator F_X that translates distorted images into the attacker's constructed image domain.

The full adversarial loss consists of a loss term from generators (F_Z, F_X) and discriminators (Q_Z, Q_X). The following equations describe the adversarial loss term.

$$L_{GAN}(F_Z, Q_Z, X_B, Z_B) = \mathbb{E}_{z_b \sim p_{enc}(z_b)}[\log Q_Z(z_b)] + \mathbb{E}_{x_b \sim p_{data}(x_b)}[\log(1 - Q_Z(F_Z(x_b)))] \tag{10}$$

where F_Z tries to generate distorted images $F_Z(x_b)$ that are similar to the real distorted images z_b, while Q_Z distinguishes between real distorted images z_b and generated distorted images $F_Z(x_b)$. F_Z minimizes the objective while Q_Z maximizes the objective, $\min_{F_Z} \max_{Q_Z} L_{GAN}(F_Z, Q_Z, X_B, Z_B)$.

$$L_{GAN}(F_X, Q_X, Z_B, X_B) = \mathbb{E}_{x_b \sim p_{data}(x_b)}[\log Q_X(x_b)] + \mathbb{E}_{z_b \sim p_{enc}(z_b)}[\log(1 - Q_X(F_X(z_b)))] \tag{11}$$

where F_X tries to generate constructed images $F_X(z_b)$ that are similar to the attacker's constructed images x_b, while Q_X distinguishes between the attacker's real constructed data and generated $F_X(z_b)$ constructed data. F_X minimizes the objective while Q_X maximizes the objective, $\min_{F_X} \max_{Q_X} L_{GAN}(F_X, Q_X, Z_B, X_B)$.

Reconstruction Cycle-GAN Cycle Consistency Loss. Next, we compute the cycle consistency loss terms using generator F_Z and generator F_X. First, the attacker translates his constructed data set X_B into the distorted image domain using generator F_Z. Then the generated distorted image is translated back into the attacker's constructed image domain using generator F_X. Second, the attacker translates distorted images Z_B into the constructed image domain using generator F_X. Then the generated construct image set is translated back into distorted image domain using generator F_Z. The mean absolute error between the constructed images and the cycled constructed images are computed. Additionally, the mean absolute error between the distorted images and the cycled distorted images are computed.

The computed cycle consistency loss values for the constructed and distorted data are summed together below.

$$L_{cyc}(F_Z, F_X) = \mathbb{E}_{x_b \sim p_{data}(x_b)}[||F_X(F_Z(x_b)) - x_b||_1] + \mathbb{E}_{z_b \sim p_{enc}(z_b)}[||F_Z(F_X(z_b)) - z_b||_1] \tag{12}$$

$F_X(F_Z(x_b))$ is the attacker's cycled constructed data and $F_Z(F_X(z_b))$ is the attacker's cycled distorted data. The error between the cycled constructed data and real constructed data is minimized. Also, the error between the cycled distorted data and real distorted data is minimized. Both values are combined to compute the total cycle consistency loss.

Reconstruction Cycle-GAN Attack Full Objective. All of the previously discussed loss terms are summed together for the full objective. The full objective for the attack consists of two adversarial loss terms and a cycle consistency loss term.

The full objective is:

$$\begin{aligned} L(F_Z, F_X, Q_X, Q_Z) = L_{GAN}(F_Z, Q_Z, X_B, Z_B) \\ + L_{GAN}(F_X, Q_X, Z_B, X_B) \\ + \lambda L_{cyc}(F_Z, F_X) \end{aligned} \tag{13}$$

where λ controls the importance of each objective. In our experiments, $\lambda = 10$. We solve the following optimization problem:

$$F_Z^*, F_X^* = \underset{F_Z, F_X}{\operatorname{argmin}} \max_{Q_Z, Q_X} L(F_Z, F_X, Q_X, Q_Z) \tag{14}$$

Reconstruction Cycle-GAN Attack Results. The reconstruction Cycle-GAN attack results demonstrate an attacker's ability to reconstruct the data owner's original image dataset using the learned mapping function F_X^* given the data owner's encoded dataset Z i.e., $F_X^*(Z)$. Generator F_X^* was optimized to translate encoded images to plain images. The translated images are expected to consist of inherent features from the distorted image domain as Cycle-GAN learns a mapping from one domain to another. Thus, we translate the data owner's encoded set to the attacker's constructed plain image domain $F_X(Z)$ to reconstruct the data owner's original image given the data owner's encoded images. The SSIM score between the reconstructed images and the original images are shown in row 5 of Table 2. We report SSIM scores using X and $F_X(Z)$ for Chest X-ray, Dermoscopy and OCT datasets. Our results demonstrate that image reconstruction exhibits poor quality given that only the encoded set and corresponding labels are available to an attacker. Consequently, given that an attacker's knowledge is restricted to $\{Z, Y\}$ it is evident that the reconstructed images consist of poor quality when compared to original private images.

6 Conclusion

We proposed a Cycle-GAN image transformation scheme that leverages autoencoder image encoding for domain translation to enhance the privacy of deep neural networks. The visible image feature information is encoded using autoencoder and Cycle-GAN to reduce the risk of information leakage. The important feature information is retained for image classification while obfuscating the sensitive image features. In this paper, we demonstrated that the proposed Cycle-GAN image encoding method successfully enhances the privacy of sensitive image data while preserving model utility with high classification accuracy. In our experiments, we evaluated the effectiveness of our Cycle-GAN encoding scheme by assessing the privacy versus model utility trade-off using classification accuracy. Additionally, we show that our proposed method is robust against reconstruction attacks when an attacker only has access to encoded data and corresponding class labels using SSIM.

References

1. Atallah, M.J., Pantazopoulos, K.N., Rice, J.R., Spafford, E.E.: Secure outsourcing of scientific computations. In: Advances in Computers, vol. 54, pp. 215–272. Elsevier (2002)
2. Yuan, X., Wang, X., Wang, C., Squicciarini, A., Ren, K.: Enabling privacy-preserving image-centric social discovery. In: Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems, ser. ICDCS 2014, pp. 198–207. IEEE Computer Society, USA (2014). https://doi.org/10.1109/ICDCS.2014.28
3. Wu, Z., Huang, Y., Wang, L., Wang, X., Tan, T.: A comprehensive study on cross-view gait based human identification with deep CNNs. IEEE Trans. Pattern Anal. Mach. Intell. **39**(2), 209–226 (2016)
4. Packhäuser, K., Gündel, S., Münster, N., Syben, C., Christlein, V., Maier, A.: Is medical chest X-ray data anonymous? arXiv preprint arXiv:2103.08562 (2021)
5. Ma, X., et al.: Understanding adversarial attacks on deep learning based medical image analysis systems. Pattern Recognit. **110**, 107332 (2021). https://doi.org/10.1016/j.patcog.2020.107332
6. Tanaka, M.: Learnable image encryption. In: 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 1–2 (2018)
7. Sirichotedumrong, W., Maekawa, T., Kinoshita, Y., Kiya, H.: Privacy-preserving deep neural networks with pixel-based image encryption considering data augmentation in the encrypted domain. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 674–678 (2019)
8. Sirichotedumrong, W., Kiya, H.: A GAN-based image transformation scheme for privacy-preserving deep neural networks (2020). https://arxiv.org/abs/2006.01342
9. Li, T., Li, N.: On the tradeoff between privacy and utility in data publishing. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 517–526 (2009)
10. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks (2020)
11. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982), pp. 160–164. IEEE (1982)
12. Chase, M., Gilad-Bachrach, R., Laine, K., Lauter, K., Rindal, P.: Private collaborative neural network learning. Cryptology ePrint Archive (2017)
13. Mohassel, P., Zhang, Y.: Secureml: a system for scalable privacy-preserving machine learning. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 19–38 (2017)
14. Wagh, S., Gupta, D., Chandran, N.: Securenn: 3-party secure computation for neural network training. Proc. Priv. Enhancing Technol. **2019**(3), 26–49 (2019)
15. Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., Taft, N.: Privacy-preserving ridge regression on hundreds of millions of records. In: 2013 IEEE Symposium on Security and Privacy, pp. 334–348 (2013)
16. Aono, Y., Hayashi, T., Trieu Phong, L., Wang, L.: Scalable and secure logistic regression via homomorphic encryption. In: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, pp. 142–144 (2016)
17. Bonte, C., Vercauteren, F.: Privacy-preserving logistic regression training. BMC Med. Genomics **11**(4), 13–21 (2018)
18. Crawford, J.L.H., Gentry, C., Halevi, S., Platt, D., Shoup, V.: Doing real work with FHE: the case of logistic regression. Cryptology ePrint Archive, Paper 2018/202 (2018). https://eprint.iacr.org/2018/202

19. Graepel, T., Lauter, K., Naehrig, M.: ML confidential: machine learning on encrypted data. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 1–21. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37682-5_1
20. Kim, M., Song, Y., Wang, S., Xia, Y., Jiang, X., et al.: Secure logistic regression based on homomorphic encryption: design and evaluation. JMIR Med. Inform. **6**(2), e8805 (2018)
21. Nandakumar, K., Ratha, N., Pankanti, S., Halevi, S.: Towards deep neural network training on encrypted data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019)
22. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: challenges, methods, and future directions. IEEE Signal Process. Mag. **37**(3), 50–60 (2020)
23. Bonawitz, K., et al.: Towards federated learning at scale: system design. Proc. Mach. Learn. Syst. **1**, 374–388 (2019)
24. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data, arXiv preprint arXiv:1806.00582 (2018)
25. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency (2016). https://arxiv.org/abs/1610.05492
26. McPherson, R., Shokri, R., Shmatikov, V.: Defeating image obfuscation with deep learning, arXiv preprint arXiv:1609.00408 (2016)
27. Huang, Y., Song, Z., Li, K., Arora, S.: InstaHide: instance-hiding schemes for private distributed learning. In: Daume III, H., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, vol. 119, pp. 4507–4518. PMLR (2020). https://proceedings.mlr.press/v119/huang20i.html
28. Yala, A., et al.: Neuracrypt: hiding private health data via random neural networks for public training (2021). https://arxiv.org/abs/2106.02484
29. Carlini, N., et al.: Is private learning possible with instance encoding? (2020). https://arxiv.org/abs/2011.05315
30. Raynal, M., Achanta, R., Humbert, M.: Image obfuscation for privacy-preserving machine learning (2020). https://arxiv.org/abs/2010.10139
31. Carlini, N., Garg, S., Jha, S., Mahloujifar, S., Mahmoody, M., Tramer, F.: Neuracrypt is not private (2021)
32. Sirichotedumrong, W., Kinoshita, Y., Kiya, H.: Pixel-based image encryption without key management for privacy-preserving deep neural networks. IEEE Access **7**, 177844–177855 (2019)
33. Chen, Z., Zhu, T., Xiong, P., Wang, C., Ren, W.: Privacy preservation for image data: a GAN-based method. Int. J. Intell. Syst. **36**(4), 1668–1685 (2021)
34. Sirichotedumrong, W., Kiya, H.: A GAN-based image transformation scheme for privacy-preserving deep neural networks (2020)
35. Kermany, D.S., et al.: Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell **172**(5), 1122–1131 (2018)
36. Scarlat, A.: dermoscopic pigmented skin lesions from ham10k (2019). https://www.kaggle.com/drscarlat/melanoma. Accessed 02 May 2020
37. Rasul, M.F., Kumar Dey, N., Hashem, M.: A comparative study of neural network architectures for lesion segmentation and melanoma detection (2020)
38. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). https://arxiv.org/abs/1512.03385
39. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: the missing ingredient for fast stylization (2017)