Privacy Enhancement for Cloud-Based Few-Shot Learning

Archit Parnami*, Muhammad Usama†, Liyue Fan‡ and Minwoo Lee§
Department of Computer Science
University of North Carolina at Charlotte, USA
Email: *aparnami@uncc.edu, †msaleem2@uncc.edu,
‡liyue.fan@uncc.edu, §minwoo.lee@uncc.edu

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892135

***Abstract*—Requiring less data for accurate models, few-shot learning has shown robustness and generality in many application domains. However, deploying few-shot models in untrusted environments may inflict privacy concerns, e.g., attacks or adversaries that may breach the privacy of user-supplied data. This paper studies the privacy enhancement for the few-shot learning in an untrusted environment, e.g., the cloud, by establishing a novel privacy-preserved embedding space that preserves the privacy of data and maintains the accuracy of the model. We examine the impact of various image privacy methods such as blurring, pixelization, Gaussian noise, and differentially private pixelization (DP-Pix) on few-shot image classification and propose a method that learns privacy-preserved representation through the joint loss. The empirical results show how privacy-performance trade-off can be negotiated for privacy-enhanced few-shot learning.**

***Index Terms*—few-shot learning, privacy, cloud, image classification, differential privacy, meta-learning**

I. Introduction

There has been a widespread adoption of cloud-based machine learning platforms recently, such as Amazon Sagemaker [1], Google AutoML [2], and Microsoft Azure [3]. They allow companies and application developers to easily build and deploy their AI applications as a Service (AIaaS).However, the users of AIaaS services may encounter two major challenges. 1) **Large Data Requirement**: Deep Learning models usually require large amounts of training data. This training data needs to be uploaded to the cloud services for the developers to build their models, which may be inconvenient and infeasible at times. 2) **Data Privacy Concerns**: Sharing data with untrusted servers may pose threats to end-user privacy. For instance, a biometric authentication application deployed in the cloud will expose user photos to a third-party cloud service.

To address the *large data requirement* problem, there has been increasing research on the approaches that require less amount of training data, popularly known as Few-Shot Learning [4]. Specifically, metric-based few-shot classification methods [5]–[9] learn to map images of unseen classes into distinct embeddings that preserve the distance relation among them and then perform classification of the input query image by the distance to the class embeddings. Recent works have been able to achieve up to ∼90% accuracy on the challenging task of 5-way 5-shot classification on the MiniImageNet dataset [10]. Despite the success and promises of few-shot learning, it is imperative to address the *data privacy concerns* to protect user-supplied sensitive data, e.g., when a metric-based few-shot model is deployed in a cloud server (Fig. 1).

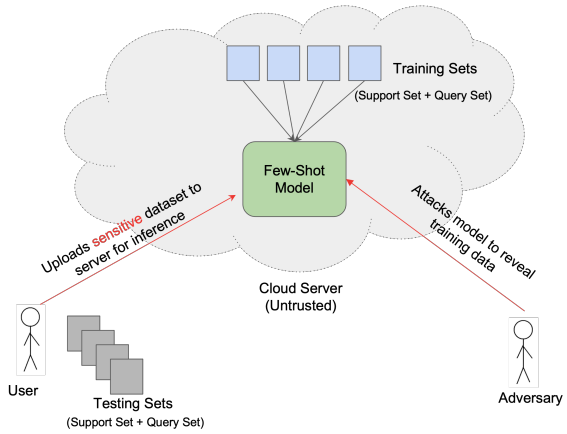


Figure 1: Threats in a cloud-based few-shot model. 1) attacks on training data [11], [12] and 2) exposure of sensitive dataset to untrusted cloud server for inference.

Several privacy-preserving approaches may be adopted in machine learning applications, including cryptography, differential privacy, and data obfuscation. Recent works [13] adopted cryptographic techniques to protect the confidentiality of data. For example, remote machine learning services can provide results on encrypted queries [13]; a range of primitives, such as Homomorphic Encryption, may be adopted to manage the encrypted data. Despite promising results, crypto-based methods inflict high computational overheads, creating challenges for practical deployment. Furthermore, such solutions may breach privacy by disclosing the exact computation results, and an adversary may utilize the model's output to launch inference attacks on training data [11], [12]. Differential privacy [14] has been adopted to train machine learning models while providing indistinguishability guarantees for individual records in the training set [15]. However, the strong privacy guarantees tend to reduce the model performance and have shown disparate impacts on the underrepresented classes [16]. In contrast, data obfuscation methods achieve privacy protection without inflicting high computational costs, e.g., image blurring and pixelization. Obfuscation can be applied to protecting both training and testing data, and can provide differential privacy guarantees at

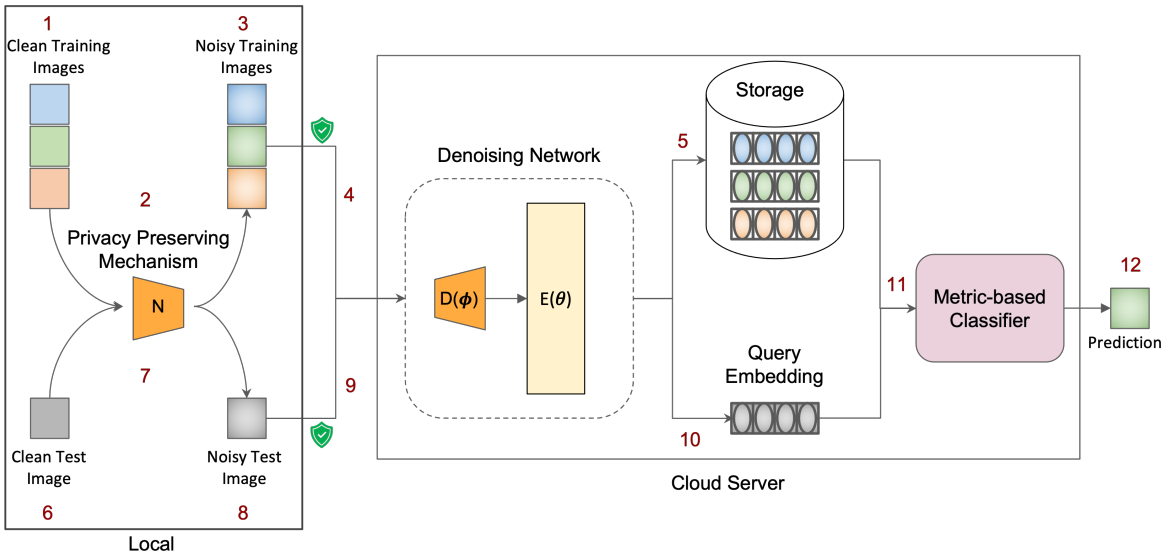


Figure 2: **Few-Shot Private Image Classification in the Cloud:** A denoising network is first trained with non-user data and deployed in the cloud. Using a privacy preserving method (2), a user can obfuscate clean training images (1) to obtain noisy training images (3). These images are then sent to the cloud server where they are first denoised and then encoded (4) to be stored as privacy-preserved embeddings on the server (5). A user can obfuscate the clean test image (6) and query the server using a noisy test image (8) to obtain a prediction (12).

individual-level data [17].

This paper focuses on the privacy of testing data (support+query) specifically for few-shot learning. A few-shot model built for clean images exhibits poor performance when tested with noisy/private image data. This is because meta-learning based few-shot models do not work well with out-of-distribution tasks [5], [18], [19]. Therefore, applying the obfuscation methods to the image data and simply using an off-the-shelf pre-trained few-shot model leads to degradation in performance, as observed in our experiments (Fig. 3 Baseline Model). Hence, it is imperative to study privacy specifically in context of few-shot learning. To this end, we suggest a private few-shot learning approach trained on noisy data samples as illustrated in Fig. 2. Adopting an obfuscation mechanism on the local input data samples, a user transfers privacy-encoded data to the cloud. The proposed jointly-trained, denoised embedding network, the *Denoising Network*, constructs privacy-preserved latent space for robust few-shot classification. To validate the proposed approach, we examine four privacy methods including traditional obfuscation methods such as Pixelization and Blurring, which do not provide quantifiable privacy guarantees [20], and also Differentially Private Pixelization (DP-Pix) [17] which provides differential privacy guarantees.

This study examines practical implications for a holistic private few-shot learning framework on an untrusted service platform, which has not been studied previously. Thus, our main contributions are 1) first proposing a unified framework for deploying few-shot learning models in the cloud while protecting the privacy of user-supplied sensitive data and 2) thoroughly examining privacy methods on three different datasets of varying difficulty, therefore 3) discovering and observing the existence of the effective privacy-preserved latent space for few-shot learning.

II. Few-Shot Learning

Few-shot learning is a subfield of machine learning that focuses on the ability of machine learning models to generalize from few-training examples. The recent progress in the field has largely come from a machine learning technique called meta-learning [18]. The idea is to train a model on numerous different but similar kinds of tasks such that the model can generalize on a new test task (as long as the test task is from the same distribution of the tasks the model was initially trained on). There are three kinds of Few-shot learning: Metric-based, optimization-based, and model-based [21]. Here we only discuss metric-based techniques and refer the reader to our survey paper [19] for further reading.

The early nominal works in metric-based few-shot learning methods are: Prototypical Networks [5], Matching Networks [6], Relation Networks [8] etc. In all these methods, the network learns to encode embeddings of input images such that images that belong to same class are closer to each other and those from different class are farther apart, where the idea of closeness is defined in terms of a metric such as euclidean.

A. Few-Shot Classification

We base our framework (Fig. 2) on Prototypical Networks [5] for building our **Few-Shot Private Image Classification (FS-PIC)** model. The model is trained on a labeled dataset D_{train} and tested on D_{test}. The set of classes present in D_{train} and D_{test} are disjoint. The test set has only a few labeled samples per class. We follow an episodic training paradigm in which each episode the model is trained to solve

an N-way K-Shot private image classification task. Each episode e is created by first sampling N classes from the training set and then sampling two sets of examples from these classes: (1) the support set $S_e = \{(s_i, y_i)\}_{i=1}^{N \times K}$ containing K examples for each of the N classes and (2) the query set $Q_e = \{(q_j, y_j)\}_{j=1}^{N \times H}$ containing H different examples from the same N classes. The episodic training for the FS-PIC task minimizes, for each episode, the loss on the prediction of samples in the query set, given the support set. The model is a parameterized function, and the loss is the negative log-likelihood of the true class of each query sample:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{|Q_e|} \log P_\theta(y_t \mid q_t, S_e), \tag{1}$$

where $(q_t, y_t) \in Q_e$ is the sampled query, S_e is the support set at episode e, and θ is the model parameter.

Prototypical Networks make use of the support set to compute a centroid (prototype) for each class (in the sampled episode) and query samples are classified based on the distance to each prototype. For instance, a CNN $f : \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_p}$, parameterized by θ_f, learns a n_p-dimensional space where n_v-dimensional input samples of the same class are close and those of different classes are far apart. For every episode e, each embedding prototype p_c (of class c) is computed by averaging the embeddings of all support samples of class c as

$$p_c = \frac{1}{K} \sum_{(s_i, y_i) \in S_e^c} f(s_i), \tag{2}$$

where $S_e^c \subset S_e$ is the subset of support examples belonging to class c. Given a distance function d, the distance of the query q_t to each of the class prototypes p_c is calculated. By taking a softmax [22] of the measured (negative) distances, the model produces a distribution over the N classes in each episode:

$$P(y = c \mid q_t, S_e, \theta) = \frac{exp(-d(f(q_t), p_c))}{\sum_n exp(-d(f(q_t), p_n))}, \tag{3}$$

where metric d is a Euclidean distance and the parameters θ of the model are updated with stochastic gradient descent by minimizing Equation (1). Once the training finishes, the parameters θ of the network are frozen. Then, given any new FS-PIC task, the class corresponding to the maximum P is the predicted class for the input query q_t.

III. Privacy Methods

We study following methods to introduce privacy in images.

A. Independent Gaussian Noise

Introducing some noise in an image is one way to distort information [23]. Kim [24], first publicized the work on additive noise by the general expression $Z = X + \epsilon$, where X is the original data point, ϵ is the random variable (noise) with a distribution $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and Z is the transformed data point, obtained by the addition of noise ϵ to the input X.

Therefore, for an image with dimensions (H, W, C), we sample $H \times W \times C$ values from a Gaussian (normal) distribution with mean (μ) zero and standard deviation σ of the probability density function $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x-\mu)^2}{2\sigma^2}$. We use the implementation from [25].

B. Common Image Obfuscation

Two widely used image obfuscation techniques are *Pixelization* and *Blurring*.

a) Pixelization [26]: (also referred to as mosaicing) can be achieved by superposing a rectangular grid of size $b \times b$ over the original image and averaging the color values of the pixels within each grid cell.

b) Blurring: i.e., Gaussian blur, removes details from an image by convolving a 2D Gaussian kernel with the image. Let the radius of blur be r, then the size of the 2D kernel is given by $(2r+1) \times (2r+1)$. Then, the values in this 2D kernel are sampled from the distribution:

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x^2 + y^2)}{2\sigma^2}, \tag{4}$$

where (x, y) are the coordinates inside the 2D kernel with origin at the center and the standard deviation σ is approximated from the radius r [27]. We use Pillow Image Library [28] for the implementation.

C. Differentially Private Image Pixelization

Differential privacy (DP) is the state-of-the-art privacy paradigm for statistical databases [14]. Differentially Private Pixelization (DP-Pix) [17] extends the DP notion to image data publication. It introduces a concept of m-Neighborhood, where two images (I_1 and I_2) are neighboring images if they differ by at most m pixels. By differential privacy, content represented by up to m pixels can be protected. A popular mechanism to achieve DP is the Laplace mechanism. However, the global sensitivity of direct image perturbation would be very high i.e., $\Delta I = 255m$, leading to high perturbation error. The DP-Pix method first performs pixelization P_b (with grid cells of $b \times b$ pixels) on the input image I, and then applies Laplace perturbation to the pixelized image $P_b(I)$, effectively reducing the sensitivity $\frac{255m}{b^2}$. The following equation summarizes the algorithm ($\tilde{P}_b$) to achieve ϵ-differential privacy:

$$\tilde{P}_b(I) = P_b(I) + L_p, \tag{5}$$

where each value in L_p is randomly drawn from a Laplace distribution with mean 0 and scale $\frac{255m}{b^2\epsilon}$. The parameter $\epsilon > 0$ specifies the level of DP guarantee, where smaller values indicate stronger privacy. As DP is resistant to post-processing [14], any computation performed on the output of DP-Pix, i.e., the perturbed pixelized images, would not affect the ϵ-DP guarantees. Our approach proposes a denoising module for the obfuscated images by DP-Pix, improving the latent representation without sacrificing DP guarantees.

IV. Privacy Enhanced Few-shot Image Classification

To build a few-shot model that can preserve the privacy of the input images, we can utilize any of the privacy methods discussed in the previous section. However, doing so may degrade the few-shot classification performance tremendously. To avoid this, we introduce a denoiser and train it jointly for few-shot classification using meta-learning on noisy images (Fig. 2). Together, the denoiser and the embedding network forms our *Denoising Network*. Combined with the properly chosen privacy method, the Denoising Network aims to discover a privacy-preserved latent embedding space (not denosing to recover the original image), where **the privacy of input data is be preserved** and **robustness and generality for few-shot classification are maintained**.

Denoiser: Zhang et al. [29] proposed a denoising convolutional neural network (DnCNN) which uses residual learning to output Gaussian noise. Specifically, the input of the network is a noisy observation such that $y = x + v$ where y is the input image, x be the clean image, and v be the actual noise. The network learns the residual mapping function $\mathcal{R}(y) \approx v$ and predicts the clean image using $x = y - \mathcal{R}(y)$. The averaged mean squared error between the predicted residue and actual noise is used as the loss function to train this denoiser with parameters ϕ as

$$\mathcal{L}(\phi) = \frac{1}{2N} \sum_{i=1}^{N} ||\mathcal{R}(y_i; \phi) - (y_i - x_i)||^2. \tag{6}$$

We plug the DnCNN denoiser into our FS-PIC pipeline (Fig. 2) to estimate the clean image before pixelization, blurring, Gaussian noise, and DP-Pix.

Embedding Network: Partially denoised images from the denoiser $D(\phi)$ are fed to embedding network $E(\theta)$ to obtain denoised embeddings, which then form the class prototypes. The classification loss is measured using Eq. 1.

The total loss for training the *Denoising Network* (Denoiser + Embedding Network) is formulated as the sum of denoising loss and classification loss:

$$\mathcal{L} = \mathcal{L}(\phi) + \mathcal{L}(\theta). \tag{7}$$

The joint loss enforces the reduction of noise in input images while learning the distinctive representations that maximize the few-shot classification accuracy. This simple loss guides the embedding space towards privacy-preserved latent space without losing its generality. For Prototypical Networks, the prototypes are expected to be the centers of the privacy-preserved embeddings for each class. Although the sum of losses can be weighted, our experiments observed that weighting did not significantly impact the final accuracy of the few-shot image classification model as long as the weighting coefficients are non-zero. We outline the episodic training process used for building a FS-PIC model in Algorithm 1 and describe the notations used in Table I.

Notation	Description
t	#examples in the training set
M	#classes in the training set
$N <= M$	#classes sampled per episode
K	#support examples sampled per class
H	#query examples sampled per class

Table I: Symbols

Algorithm 1: FS-PIC model training

Input: $D = \{(x_1, y_1), ..., (x_t, y_t)\}$ where $y_i \in \{1, ..., M\}$. D^c denotes the subset of D containing all elements (x_i, y_i) such that $y_i = c$.

while *True* **do**
- `// Select a set of N classes`
- $V \leftarrow$ RandomSample$(\{1, ..., M\}, N)$
- **for** c *in* V **do**
  - `// Select support examples`
  - $S_e^c \leftarrow$ RandomSample(D^c, K)
  - `// Select query examples`
  - $Q_e^c \leftarrow$ RandomSample$(D^c \setminus S_e^c, H)$
  - `// Add noise`
  - $\hat{S}_e^c \leftarrow$ AddNoise(S_e^c, ϵ)
  - $\hat{Q}_e^c \leftarrow$ AddNoise(Q_e^c, ϵ)
- **end**
- `// Form a set of all clean images`
- $S_e \leftarrow \{S_e^1, S_e^2, ...S_e^N\}$
- $Q_e \leftarrow \{Q_e^1, Q_e^2, ...Q_e^N\}$
- $X_e \leftarrow \{S_e, Q_e\}$
- `// Form a set of all noisy images`
- $\hat{S}_e \leftarrow \{\hat{S}_e^1, \hat{S}_e^2, ...\hat{S}_e^N\}$
- $\hat{Q}_e \leftarrow \{\hat{Q}_e^1, \hat{Q}_e^2, ...\hat{Q}_e^N\}$
- $\hat{X}_e \leftarrow \{\hat{S}_e, \hat{Q}_e\}$
- `// Apply the denoiser`
- $\bar{X}_e \leftarrow G(\hat{X}_e; \theta)$
- $\bar{S}_e, \bar{Q}_e \leftarrow \bar{X}_e$
- `// Calculate denoising loss`
- $\mathcal{L}_d \leftarrow$ MSE$(\bar{X}_e, X_e)$
- `// Compute class prototypes using denoised support examples`
- **for** c *in* V **do**
  - $\bar{p_c} \leftarrow \frac{1}{K} \sum_{(\bar{x_i}, y_i) \in \bar{S_e^c}} f_\phi(\bar{x_i})$
- **end**
- $\mathcal{L}_c \leftarrow 0$
- **for** c *in* V **do**
  - **for** $(\bar{x_i}, y_i)$ *in* $\bar{Q_e^c}$ **do**
    - $\mathcal{L}_c \leftarrow \mathcal{L}_c + \frac{1}{NH}[d(f_\phi(\bar{x_i}), \bar{p_c}) + \log \sum_{c'} \exp(-d(f_\phi(\bar{x_i}), \bar{p_c}))]$
  - **end**
- **end**
- $\mathcal{L} \leftarrow \mathcal{L}_d + \mathcal{L}_c$
- $\phi \leftarrow \phi - \alpha_\phi \frac{\partial \mathcal{L}}{\partial \phi}$
- $\theta \leftarrow \theta - \alpha_\theta \frac{\partial \mathcal{L}}{\partial \theta}$

end

V. Experiments

Datasets: 1) *Omniglot* [30] is a dataset of 1623 handwritten characters collected from 50 alphabets. Each character has 20 examples drawn by a different human subject. We follow the same procedure as in [6] by resizing the gray-scale images to 28×28 and augmenting the character classes with rotations in multiples of 90 degrees. Our training, validation, and testing split is of sizes 1028, 172, and 423 characters, respectively (or $4\times$ with augmentation). **2)** *CelebFaces Attributes Dataset (CelebA)* [31] is a large-scale face attributes dataset with more than 10K celebrity (classes) images. For the purpose of our experiments, we select classes that have at least 30 samples. This gives us 2360 classes in total, out of which 1510 are used for training, 378 for validation, and 427 for testing. We use aligned and cropped version of the dataset in which images are of dimension $218(h) \times 178(w)$. We center crop each image to 176×176 and then resize to 84×84. **3)** *MiniImageNet* [6] dataset contains 100 general object classes where each class has 600 color images. The images are resized to 84×84, and the dataset is split into 64 training, 16 validation, and 20 testing classes following [5].

Settings for Privacy Methods: We explore the following parameters for each privacy method. Gaussian Blur with radius $r = \{1, 2, 3, 4, 5\}$ is used for blurring images. A filter window of size $b \times b$ where $b = \{2, 4, 6, 8, 10\}$ is used for pixelization. The pixelated image is then resized to match the model input dimensions. We perform experiments with Gaussian noise $\epsilon \sim \mathcal{N}(\mu, \sigma)$ with mean μ = 0 and standard deviation $\sigma = \{40, 80, 120, 160, 200\}$. For DP-Pix, we fix $\epsilon = 3$, $m = 1$ and vary pixelization parameter b with values $\{2, 4, 6, 8, 10\}$.

Denoising Network: We use a lighter version of the DnCNN [29] model i.e., with 8 CNN layers instead of 17, for first denoising the image and subsequently feeding the denoised image into one of the following embedding networks. **Conv-4** is a 4-layered convolutional neural network with 64 filters in each layer originally proposed in [5] for few-shot classification. **ResNet-12** is a 12-layer CNN with 4 residual blocks. It has been shown to have better classification accuracy on few-shot image classification tasks.

Training and Evaluation: We train using N-way K-shot PIC tasks (Algorithm. 1) and use Adam optimizer with learning rate $\alpha_\theta = \alpha_\phi = 0.001$ with a decay of 0.5 every 20 epochs. Table II lists the hyperparameters for the three datasets. The network is trained to minimize total loss of denoiser and classifier (Eq. 7). We evaluate the performance by sampling 5-way 5-shot PIC tasks (with same privacy settings) from the test sets and measure the classification accuracy. The final results report the performance averaged over 1000 test episodes for the Omniglot dataset, and 600 test episodes for both MiniImageNet and CelebA datasets. To measure the effectiveness of the proposed denoising embedding space, we both train and evaluate each model's performance in two settings: 1) **without using the denoiser** and 2) **jointly training the denoiser with the classifier** i.e., the proposed *Denoising Network*.

Privacy Risk Evaluation: Privacy attacks on trained models such as model inversion [11] and membership inference [12] are not applicable in our setting because the denoising and embedding models are trained with publicly available classes (data) using meta-learning. The user-supplied test data (support and query set) are obfuscated for privacy protection. A practical privacy attack on obfuscated images is to infer the identities using existing facial recognition systems and public APIs, e.g., Rekognition. In this study, our goal is to investigate (1) the efficacy of the studied image obfuscation methods for privacy protection and (2) whether the proposed denoising approach has effects on privacy. To simulate a powerful adversary, we apply the state-of-the-art face recognition techniques, e.g., FaceNet with the Inception ResNet V1 network [32], on the CelebA dataset; MTCNN [33] is applied to detect and resize the facial region in each input image. Specifically, 1000 entities were randomly selected from the CelebA dataset. For each entity, we randomly sampled 30 images, which were then partitioned between training and testing (20 : 10). Different versions of the test set were generated by applying image obfuscation methods with various parameter values (denoted as `Noisy`) and by applying the proposed Denoising Network (denoted as `Denoised`). We fine-tuned the Inception network and trained an SVC classifier on the clean training data. In Fig. 5, we report the accuracy on the noisy and denoised test sets, i.e., success of re-identification, with higher values indicating higher privacy risks.

	Omniglot	CelebA	MiniImageNet
Way	60	5	5
Shots	5	5	5
Query	5	5	15
Epochs	500	200	200
Patience	50	20	20
Episodes	100	100	100

Table II: Training Hyperparameters

VI. Results and Discussions

	Omniglot	CelebA	MiniImageNet
Conv-4	0.99	0.90	0.61
ResNet-12	–	0.92	0.65

Table III: Baseline test accuracy of 5-way 5-shot classification of clean images. Omniglot is not evaluated for ResNet-12 because of its already near 100% performance.

A. Task Difficulty

The average 5-way 5-shot classification accuracy of our baseline few-shot model [5] trained on clean images and tested on clean images is 99% on Omniglot dataset, 91% on CelebA dataset, and 61% on MiniImageNet dataset using Conv-4 encoder (Table III). This shows the approximate level of difficulty of few-shot tasks for each dataset i.e., Omniglot tasks are easy, tasks from CelebA have medium difficulty, and MiniImageNet tasks are hard.

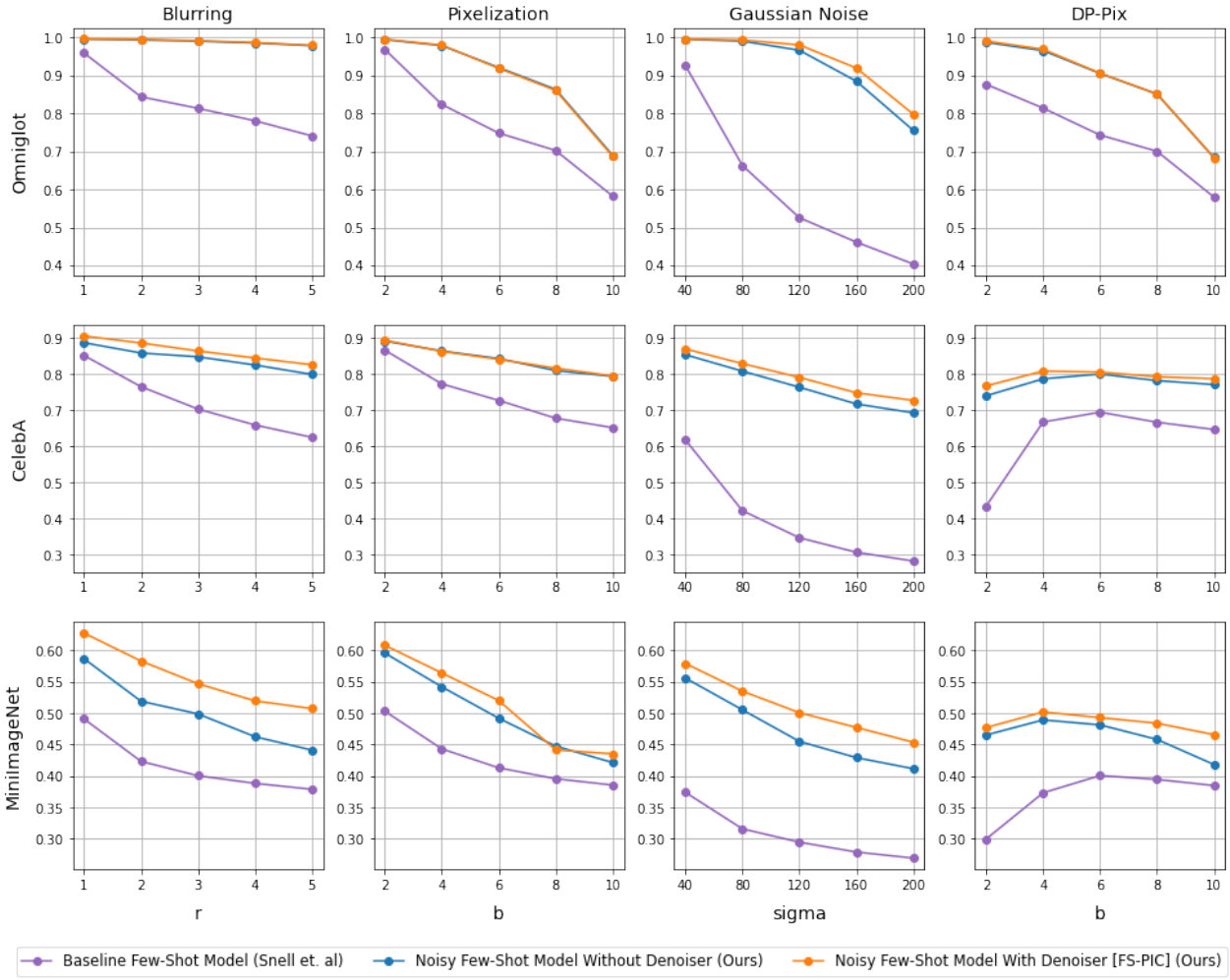


Figure 3: Test accuracy (y-axis) of 5-way 5-shot private image classification tasks sampled from Omniglot (top), CelebA (center) and MiniImageNet (bottom) datasets, presented with different privacy settings (x-axis) when using Conv-4 as encoder.

B. Generalization

We compare results for few-shot private image classification using three models in Fig. 3:

1) **Baseline Few-Shot Model:** When the few-shot model is trained on clean images and is tested on noisy images.
2) **Noisy Few-Shot Model Without Denoiser:** When the baseline few-shot model is trained on noisy images and is tested on noisy images with same privacy settings.
3) **Noisy Few-Shot Model With Denoiser:** When the baseline few-shot model is *jointly trained with the denoiser* on noisy images and is tested on noisy images with same privacy settings (Algorithm 1).

In all cases, we observe that noisy few-shot models outperforms the baseline few-shot model with wide gap. Also, in most cases, we note that adding a denoiser improves the accuracy. To better observe the effectiveness of denoiser, in Fig. 4, we quantify the improvement by calculating $\%\ \text{Gain} = \frac{\text{accuracy with denoiser} - \text{accuracy without denoiser}}{\text{accuracy without denoiser}} \times 100$. We also quantify the change to the original image caused by the privacy method (post denoising) by calculating Structural Similarity Index (SSIM) [34] between denoised image and original clean image, averaged over 100 test images for each dataset and privacy parameter.

Blurring, Pixelization and Gaussian Noise: As we increase the value of privacy parameters, the SSIM decreases, suggesting the higher dissimilarity between the denoised images and the original image (Fig 4). Despite the degradation caused by the privacy method to the original image, we observe positive % Gains for all three datasets. Specifically, on hard tasks (MiniImageNet), a gain of upto 15% in accuracy ($r = 5$) with the proposed *Denoising Network*, reaffirming the generality of the few-shot learning. For easy (Omniglot) and medium (CelebA) tasks, where the baseline accuracy is already high, a relatively small positive gain of up to 5% ($\sigma = 200$) is reported.

DP-Pix: As we increase the size of pixelization window (b), the amount of Laplace noise that we add to the image decreases (as defined by $\frac{255m}{b^2\epsilon}$); however, the image quality

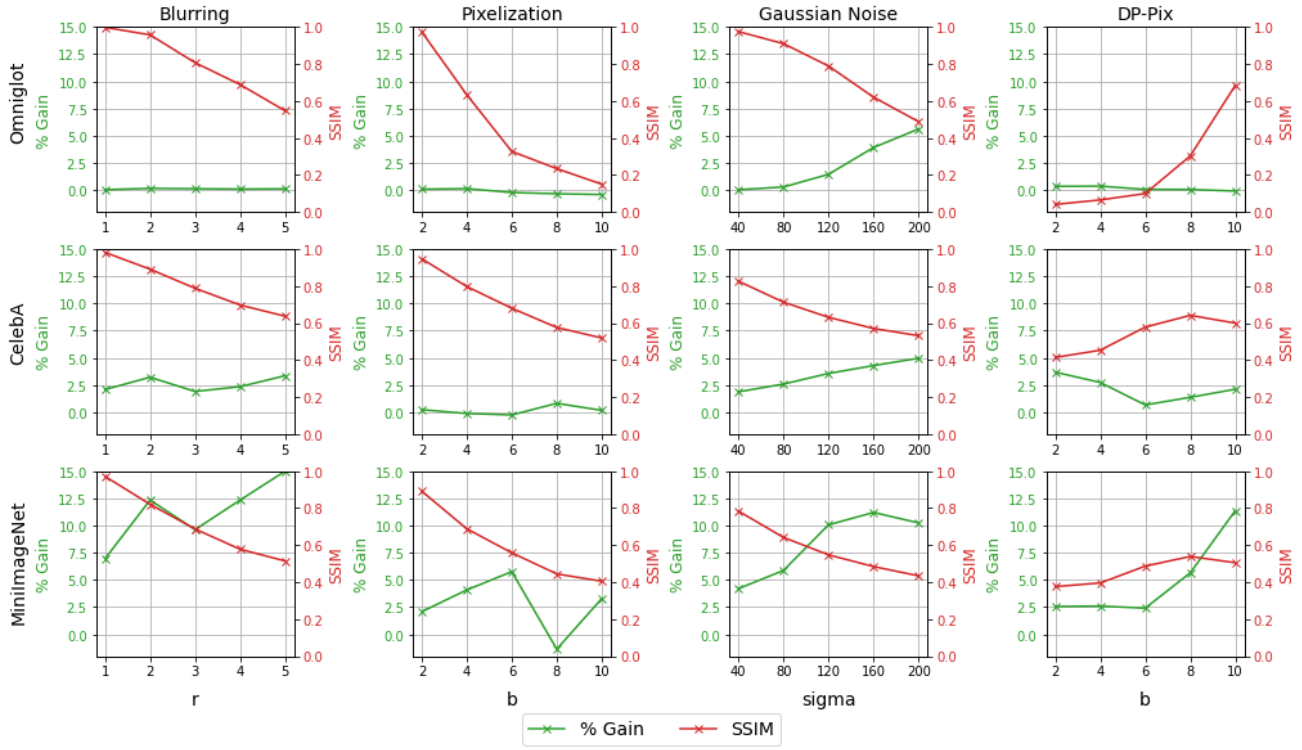


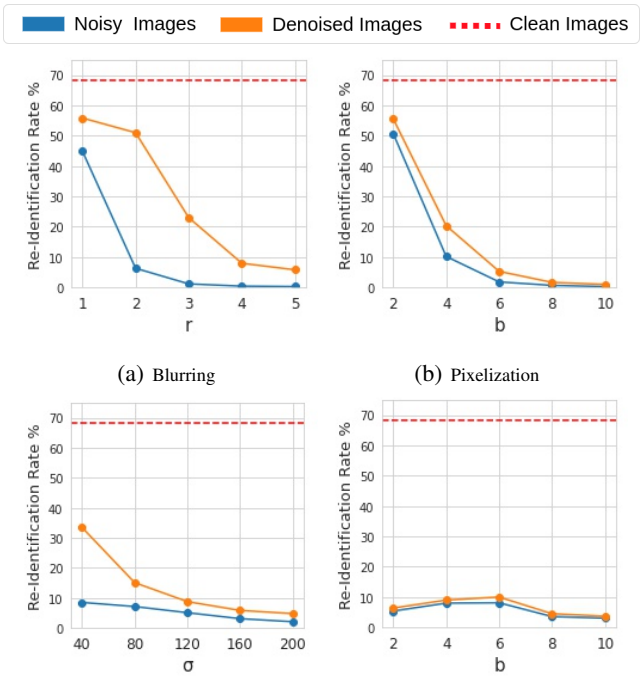
Figure 4: % Gain vs SSIM for Conv-4

decreases because of increasing pixelization. Therefore, we observe a trade-off point where the accuracy first increases and then decreases as we increase b (Fig 3). This trade-off is particularly observed for CelebA and MiniImageNet datasets. For Omniglot dataset, the performance just decreases with increasing b because of the low resolution images in the dataset. From Fig. 4, we observe that DP-Pix has the lowest SSIM values when compared with other privacy methods causing the most notable changes on the original image. Interestingly, we note that even with low SSIM values, we find instances that exhibit moderate % gain i.e., at $b = 2, 4, 6$ indicating the presence of privacy preserving denoising embeddings.

C. Empirical Privacy Risks

As described earlier, this experiment showcases the efficacy of image obfuscation methods against a practical adversary who utilizes state-of-the-art face recognition models trained on clean images. Furthermore, this experiment investigates whether the proposed denoising network has effects on privacy protection empirically. To this end, we vary the algorithmic parameters of the privacy methods and report the face re-identification rates on both obfuscated and denoised images in Fig. 5. For the clean test set sampled from CelebA, the re-identification accuracy is 68.12%.

Blurring: In Fig. 5a, the privacy risks are quite high with Gaussian kernel size $r = 1$ for both blurred and denoised images. As we increase the radius r, the chance of re-identifying the image decreases rapidly. We also observe that after denoising, the blurred images are more likely to be re-identified. For instance, at $r = 2$, the re-identification rate is 7.34% for blurred images but 54.01% for denoised images.



(a) Blurring (b) Pixelization

(c) Additive Gaussian (d) DP-Pix ($\epsilon = 3$)

Figure 5: Privacy Risk Evaluation with CelebA

Pixelization: As shown in Fig. 5b, small cell size in pixelization, e.g., $b = 2$, leads to high face re-identification

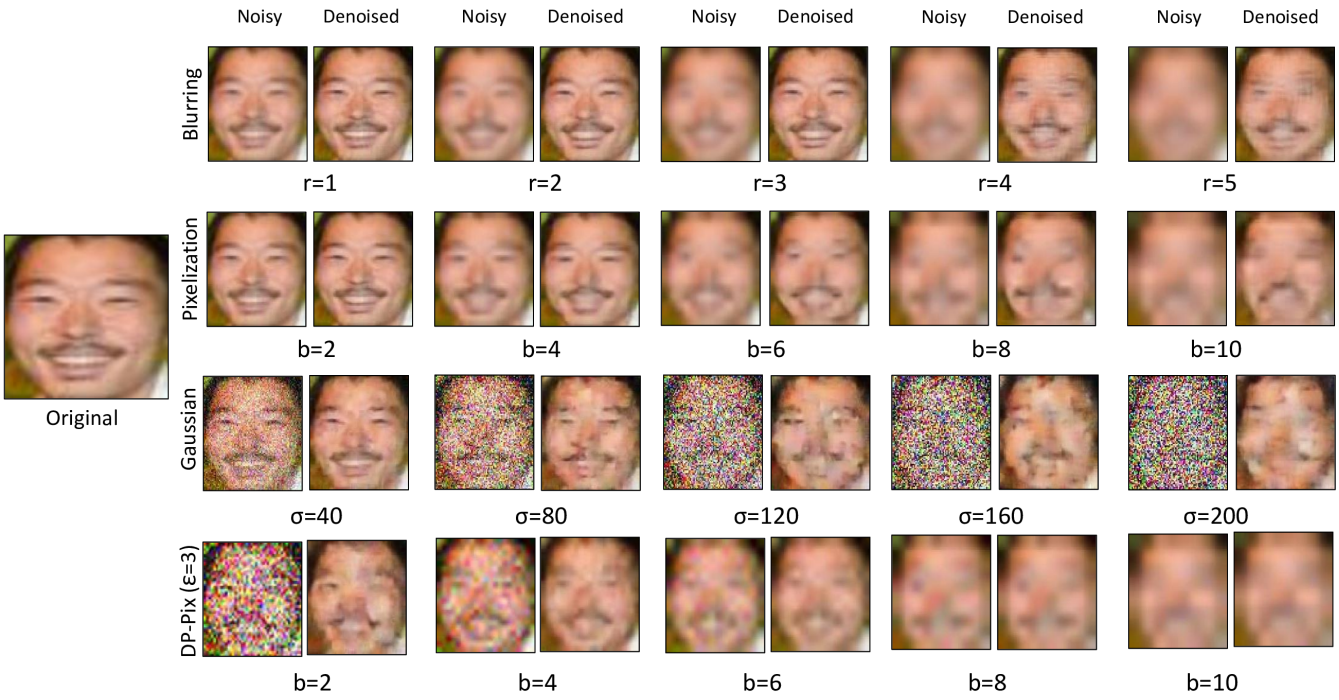


Figure 6: **Qualitative Evaluation of Privacy Methods:** The figure depicts the obfuscated images (`Noisy`) by the studied privacy methods at various parameters, as well as the denoised output, with a sample input from CelebA.

rates for both pixelized and denoised images. Increasing b helps reduce the rate of face re-identification rapidly, e.g., from 53.73% to 4.82% for pixelized images by increasing b from 2 to 6. Denoising slightly increases the privacy risk, but the additional risk diminishes with larger b values and is much lower than observed in blurring.

Additive Gaussian: Over the range of σ values studied in our experiments, Additive Gaussian inflicts lower privacy risks with a small noise ($\sigma = 40$), compared to Blurring and Pixelization. As shown in Fig. 5c, increasing σ leads to a moderate reduction in the privacy risk. For example, face re-identification rate is 5.11% at mid noise level ($\sigma = 120$), reduced from 8.54% at low noise level ($\sigma = 40$). Denoising the obfuscated images leads to a significant increase in the privacy risk at low σ values, e.g., 33.55% higher in face re-identification rate when $\sigma = 40$.

DP-Pix Fig. 5d presents the re-identification results for images obfuscated with DP-Pix as well as those denoised by our proposed approach. We observe low privacy risks across all b values. Furthermore, performing denoising on DP-Pix obfuscated images does not lead to significant higher privacy risks with any b value, as opposed to other image obfuscation methods. While face re-identification rates are consistently low, higher rates occur when $b = 4$ and 6. Recall that higher utility was observed when $b = 4$ and 6 in Fig. 3. It has been reported in [17] that the quality of obfuscated images may be optimized by tuning b value given the privacy requirement ϵ, by balancing the approximation error by pixelization and the Laplace noise with scale $\frac{255m}{b^2\epsilon}$.

D. Qualitative Evaluation of Privacy Methods

Fig. 6 provides a qualitative evaluation on the obfuscated and denoised images generated for a range of parameter values. MTCNN [33] was applied to a sample input image of CelebA, to detect the facial region. Perceptually, the proposed denoiser may improve the image quality upon the obfuscated images to various extents. However, image quality does not always correlate with empirical privacy risks, i.e., face re-identification with public models. In combination with Fig. 5, we observe that the proposed denoising leads to various levels of privacy risk increment, while producing higher quality images. For example, the results show higher privacy risk increment for Blurring with $r = 3$ (23.33%), moderate increment for Pixelization $b = 4$ (10.14%) and Additive Gaussian $\sigma = 80$ (7.96%), and little increment for DP-Pix $b = 2$. Fig. 6 confirms that the denoiser performance may vary depending on the image obfuscation method, and that DP-Pix provides consistent privacy protection even with denoising.

E. Observation of the Privacy-Preserved Embedding Space

Fig. 7 shows the evolution of the embeddings in the process of privacy encoding and privacy-preserved representation learning by presenting the t-SNE [35] visualization of the clean, noisy, and denoised embeddings of randomly sampled 100 test images from a total of 5 classes from the CelebA

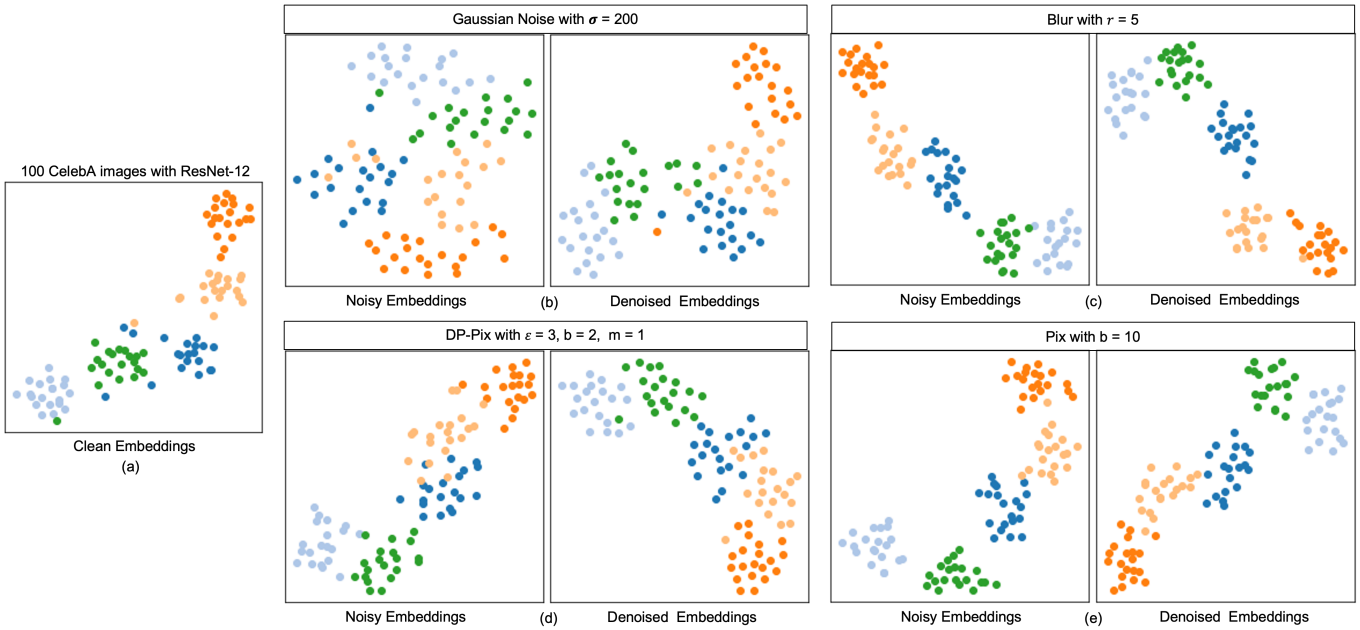


Figure 7: t-SNE Visualization

dataset. The embeddings are obtained from the ResNet-12 encoder trained under different noise settings for 5-way 5-shot classification. We say the embeddings are clean when the input images have no noise and the encoder is trained for few-shot classification of clean images. The noisy embeddings are obtained by using the encoder trained for few-shot classification of noisy images and without using the denoiser. The denoised embeddings are obtained by the proposed *Denoising Network* (Fig. 2) i.e., the encoder trained in conjunction with denoiser for few-shot classification on noisy images.

We report the results for a case when a few-shot method such as Prototypical Networks can generate good clusters for the clean images (Fig. 7a), and observe the impact on clustering with noisy images and subsequently when those images are denoised. We notice that when the initial clusters are good, pixelization (Fig. 7c) and blurring (Fig. 7e) will have little impact on the quality of the clusters even with the high amount of noise. Therefore, pixelization and blurring maintain generality (robust to noise) and are also vulnerable to re-identification. Gaussian noise (Fig. 7b) distorts the initial clusters more significantly, which can lead to lower few-shot classification performance. Applying denoising to Gaussian noise improves the clustering results, however still poses moderate privacy threat as seen in re-identification experiments (Fig. 5c). Similarly, with DP-Pix (Fig. 7d), the original clusters are also distorted upon obfuscation. But, when denoised with proposed Denoising Network, we can observe better clustering performance. Because of DP-Pix's privacy guarantee and lowest re-identification rates, we can say that the obtained denoised embeddings are privacy-protected i.e., the network finds the privacy-preserved embedding space which maintains generality (robust to noise) and also preserves privacy.

VII. Related Works

Xie et al. [36] incorporate differential privacy into few-shot learning through adding Gaussian noise into the model training process [15] to protect the privacy of training data. [37], [38] have also provided a strong privacy protection guarantee in pairwise learning for training data. On the other hand, [39] propose to use hashing to store the embedding of the input images. Similar to cryptographic approaches [13], the work [39] incurs high computational complexity to achieve accuracy. Differently, our approach addresses the privacy of user data at source (i.e., the images are already privatized before the server sees them) with strong privacy protection. To the best of our knowledge, ours is the only approach that addresses privacy in the context of few-shot metric learning for user-supplied training and testing data.

VIII. Conclusion & Future Work

In this paper, we present a novel framework for training a few-shot private image classification model, which aims to preserve the privacy of user-supplied training and testing data. The framework makes it possible to deploy few-shot models in the cloud without compromising users' data privacy. We discuss and confirm that there exists a privacy-preserved embedding space which has both stronger privacy and generalization performance on few-shot classification. The proposed method provides privacy guarantees while preventing severe degradation of the accuracy as confirmed by results on three different datasets of varying difficulty with several privacy methods. Evaluation with re-identification attacks verifies the low empirical privacy risk of our proposed method, especially with DP-Pix. While our study focuses on well-known image obfuscation methods, future research may explore scenarios

where users could apply novel image obfuscation methods locally, i.e., different from those applied to training data. Furthermore, our results motivate the future direction of searching for a more effective privacy-preserved space for few-shot learning in other domains such as speech [40]. Examination of other evaluation metrics for privacy-preserved embedding space will promote the relevant future study. We release the code for our experiments at https://github.com/ArchitParnami/Few-Shot-Privacy.

ACKNOWLEDGEMENT

This work has been supported in part by the National Science Foundation CNS-1949217, CNS-2003198 and UNC Charlotte. The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] A. V. Joshi, "Amazon's machine learning toolkit: Sagemaker," in *Machine Learning and Artificial Intelligence*, pp. 233–243, Springer, 2020.

[2] E. Bisong, "Google automl: cloud vision," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 581–598, Springer, 2019.

[3] A. Team, "Azureml: Anatomy of a machine learning service," in *Conference on Predictive APIs and Apps*, pp. 1–13, PMLR, 2016.

[4] Y. Wang and Q. Yao, "Few-shot learning: A survey," *ArXiv*, vol. abs/1904.05046, 2019.

[5] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical Networks for Few-shot Learning," *arXiv:1703.05175 [cs, stat]*, Mar. 2017. arXiv: 1703.05175.

[6] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," *arXiv:1606.04080 [cs, stat]*, June 2016. arXiv: 1606.04080.

[7] B. N. Oreshkin, P. Rodriguez, and A. Lacoste, "TADAM: Task dependent adaptive metric for improved few-shot learning," *arXiv:1805.10123 [cs, stat]*, May 2018. arXiv: 1805.10123.

[8] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to Compare: Relation Network for Few-Shot Learning," *Computer Vision and Pattern Recognition*, Nov. 2017. arXiv: 1711.06025.

[9] S. W. Yoon, J. Seo, and J. Moon, "TapNet: Neural Network Augmented with Task-Adaptive Projection for Few-Shot Learning," *International Conference on Machine Learning*, p. 9, 2019.

[10] "Few-shot image classification on mini-imagenet 5-way (5-shot)," 2021. Available at https://paperswithcode.com/sota/few-shot-image-classification-on-mini-3.

[11] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

[12] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18, IEEE, 2017.

[13] J. Cabrero-Holgueras and S. Pastrana, "Sok: Privacy-preserving computation techniques for deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2021, no. 4, pp. 139–162, 2021.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, p. 211–407, aug 2014.

[15] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, pp. 308–318, 2016. arXiv: 1607.00133.

[16] E. Bagdasaryan and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," 2019.

[17] L. Fan, "Image pixelization with differential privacy," in *Data and Applications Security and Privacy XXXII*, pp. 148–162, Springer International Publishing, 2018.

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 1126–1135, JMLR.org, 2017.

[19] A. Parnami and M. Lee, "Learning from few examples: A summary of approaches to few-shot learning," *arXiv preprint arXiv:2203.04291*, 2022.

[20] R. McPherson, R. Shokri, and V. Shmatikov, "Defeating image obfuscation with deep learning," *ArXiv*, vol. abs/1609.00408, 2016.

[21] O. Vinyals, "Model vs optimization meta learning," *Meta-Learning Symposium at Neural Information Processing Systems*, 2017. Available at http://metalearning-symposium.ml/files/vinyals.pdf.

[22] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*, pp. 227–236, Springer, 1990.

[23] K. Mivule, "Utilizing noise addition for data privacy, an overview," *ArXiv*, vol. abs/1309.3958, 2013.

[24] J.-J. Kim, "A method for limiting disclosure in microdata based on random noise and transformation," 2002.

[25] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020. Available at https://doi.org/10.1038/s41586-020-2649-2.

[26] S. Hill, Z. Zhou, L. Saul, and H. Shacham, "On the (in)effectiveness of mosaicing and blurring as tools for document redaction," in *Proceedings on Privacy Enhancing Technologies*, vol. 2016, pp. 403 – 417, 2016.

[27] P. Gwosdek, S. Grewenig, A. Bruhn, and J. Weickert, "Theoretical foundations of gaussian convolution by extended box filtering," in *Scale Space and Variational Methods in Computer Vision*, (Berlin, Heidelberg), pp. 447–458, Springer Berlin Heidelberg, 2012.

[28] A. Clark, "Pillow (pil fork) documentation," 2015. Available at https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

[29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, pp. 3142–3155, July 2017. arXiv: 1608.03981.

[30] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 33, 2011.

[31] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CoRR*, vol. abs/1503.03832, 2015.

[33] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," *CoRR*, vol. abs/1604.02878, 2016.

[34] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[35] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[36] Y. Xie, H. Wang, B. Yu, and C. Zhang, "Secure collaborative few-shot learning," *Knowl. Based Syst.*, vol. 203, p. 106157, 2020.

[37] Z. Xue, S. Yang, M. Huai, and D. Wang, "Differentially private pairwise learning revisited," in *Proceedings of International Joint Conference on Artificial Intelligence*, 2021.

[38] M. Huai, D. Wang, C. Miao, J. Xu, and A. Zhang, "Pairwise learning with differential privacy guarantees," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[39] R. Gelbhart and B. I. P. Rubinstein, "Discrete few-shot learning for pan privacy," *ArXiv*, vol. abs/2006.13120, 2020.

[40] A. Parnami and M. Lee, "Few-shot keyword spotting with prototypical networks," *arXiv preprint arXiv:2007.14463*, 2020.