Poster: On the Feasibility of Training Neural Networks with Visibly Watermarked Dataset

Sanghyun Hong*, Tae-hoon Kim†, Tudor Dumitraș*, and Jonghyun Choi‡
*University of Maryland, College Park
†Deeping Source Inc.
‡Gwangju Institute of Science and Technology (GIST)

Abstract—**As there are increasing needs of sharing data for machine learning, there is growing attention for the owners of the data to claim the ownership. Visible watermarking has been an effective way to claim the ownership of visual data, yet the visibly watermarked images are not regarded as a primary source for learning visual recognition models due to the lost visual information by in the watermark and the possibility of an attack to remove the watermarks. To make the watermarked images better suited for machine learning with less risk of removal, we propose DeepStamp, a watermarking framework that, given a watermarking image and a trained network for image classification, learns to synthesize a watermarked image that are human-perceptible, robust to removals, and able to be used as training images for classification with minimal accuracy loss. To achieve the goal, we employ the generative multi-adversarial network (GMAN). In experiments with CIFAR10, we show that the DeepStamp learn to transform a watermark to be embedded in each image and the watermarked images can be used to train networks.**

I. Introduction

Convolutional neural networks (CNN) have achieved super-human performances in various computer vision and machine learning applications with the help of large-scale supervised datasets. However, the data curation is very expensive in terms of time, fund and much effort to control the quality of supervision. To ease the cost of data curation, one can outsource the collection of a dataset to multiple parties with reasonable rewards. However, there is a risk of the shared data could be stolen and shared without any rewards to the parties. To prevent the case, we should be able to claim the ownership of the data when the data is stolen.

One possible approach is to use cryptographic methods such as homomorphic encryption [3] and multi-party computation (MPC) [5]. Homomorphic encryption allows computation on encrypted images and returns results when decrypted, matches the result of the operations as if they had been performed on the plain-data. MPC jointly computes a neural network over multiple inputs while keeping each input private. However, the computation on the encrypted data takes longer than that on the plain data. Also, once the content of the data is revealed, these schemes cannot protect the ownership since they are only designed to protect the privacy without sharing. Thus, they cannot prevent the fake ownership issues.

Another approach is to conceal secrets (messages) within data, when the data is shared with others, such as steganography or invisible watermarking [4]. Unlike the cryptographic solutions, these methods do not introduce extra computational costs because they compute on the plain data. Nevertheless, the secrets are easy to be destroyed by the data modifications, e.g., cropping, rotating, or resizing, easily exploited by an attacker. Given that the modifications are commonly used in training CNNs for the data augmentation, if an attacker modifies the data before claiming the fake ownership, the owner is hard to prove whether the data is hers or not.

We consider *visible watermarking* [1] to protect the ownership of shared data for readiness of information for claiming the ownership. There have been challenges in using the technique to the datasets for training CNNs because networks trained with watermarked data can suffer from accuracy losses, and an attacker can use sophisticated methods [2] to remove watermarks from data. Our work first studies the accuracy drops to the changes in visibility and randomness[1] of the watermarks in data. We, then, propose DeepStamp that synthesizes watermarks, once blended to data, minimizes the accuracy drop and makes them hard to be removed, with given data and a watermark image. We leverage a generative network, GMAN, to achieve randomness and implement necessary conditions as discriminators. With CIFAR10, we show that our watermarks minimize accuracy loss and, once we have watermarks, they can be used to train multiple CNNs.

II. Threat Model

We consider an adversary who *claims the ownership* of the datasets produced or collected by others such as industrial partners or public sources. For instance, suppose that Alice wants to provide a data collection A to Bob who wants to train CNNs using A as a subset of their training data. However, Alice still wants to claim the ownership of the shared data A, to prevent the case that Bob turns into malicious and takes benefit from re-sharing/selling A to other parties. Alice also can claim the ownership of A when Bob did data modifications.

III. DeepStampFramework

We aims to learn transformations of a given watermark to each data that can: (1) minimize the accuracy drop of networks trained on the watermarked data, (2) make the watermarks embedded in each data hard to remove by an attacker, and (3) make watermarks clearly perceptible to human-eyes. We illustrate the overview of our framework in Fig. 1. First, since [2] revisits that the random perturbation of watermarks embedded to data makes the watermark robust to removals, we learn the random perturbation for each data automatically during training by employing a generative network, GMAN, that enables a watermarking network W. To minimize the accuracy loss, we utilize a pre-trained CNN model F. We use an auto-encoder V and a discriminator D to enforce the transformations from W are visually similar to the original watermark at a certain level.

In training, DeepStamp minimizes:

- $\mathcal{L}_f(x, x'_w)$: the task loss computes the difference in the inferred labels of the clean data x and the same data with

[1] [2] shows that the random perturbation of watermark images embedded in data makes an attacker hard to remove the watermarks.

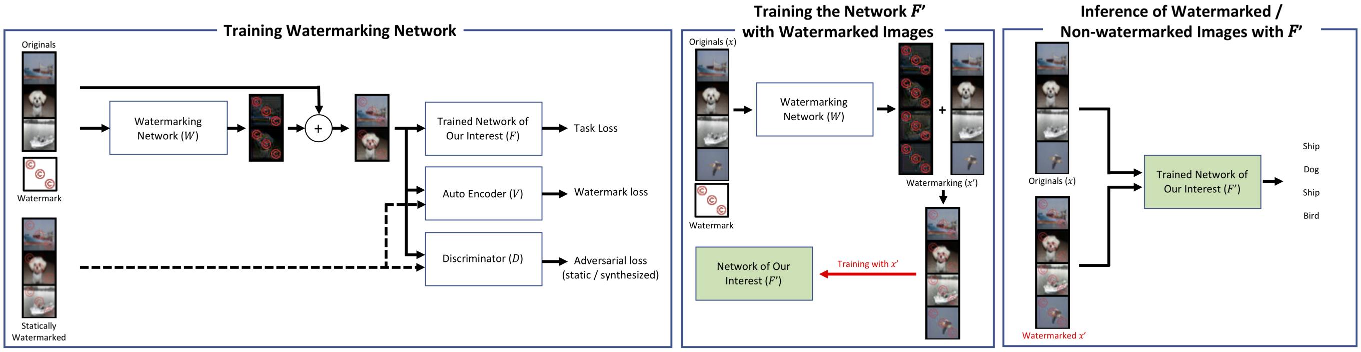

Training Watermarking Network
Originals
Watermark
Statically Watermarked
Watermarking Network (W)
+
Trained Network of Our Interest (F)
Task Loss
Auto Encoder (V)
Watermark loss
Discriminator (D)
Adversarial loss (static / synthesized)
Training the Network F' with Watermarked Images
Originals (x)
Watermarking Network (W)
Watermark
Watermarking (x')
Network of Our Interest (F')
Training with x'
Inference of Watermarked / Non-watermarked Images with F'
Originals (x)
Watermarked x'
Trained Network of Our Interest (F')
Ship
Dog
Ship
Bird

Fig. 1. **Overview of Our Watermarking Framework.** We employ Generative Multi-Adversarial Networks (GMAN) to train the watermarking netwotk (W) which returns task-aware watermark images w for each corresponding original data (x). We embed w to x and train the network of our interest (F') with the watermarked images (x'). Once the network is trained, the network (F') infers the correct labels for both x and x'.

synthesized watermarks x'_w.

- $\mathcal{L}_v(w, w')$: the ℓ_2 loss from the auto-encoder V between the original watermark w and the synthesized one w'.
- $\mathcal{L}_d(x_w, x'_w)$: the discriminator loss (binary cross entropy) that separates the original watermarked data x_w from the data with synthesized watermarks x'_w.

Thus, the entire loss that we minimize is:

$$\mathcal{L}_{tot} = \mathcal{L}_f(x, x'_w) + \mathcal{L}_v(w, w') + \mathcal{L}_d(x_w, x'_w). \qquad (1)$$

In stamping, DeepStamp synthesizes the watermarked data x'_w for a given data x and an watermark image w that is visible and hard to remove with minimal accuracy drop. Alice now can share the watermarked data x'_w to Bob, and Bob can train his network F' with x'_w. Alice does not concern about the ownership issue since w' is visible, and w' is hard to be removed by anyone else because each w' has randomly perturbed by W. Once F' trained, F' ensures the similar accuracy on both the x and watermarked data x'_w.

IV. Evaluation

Experimental Setup. We use CIFAR10 that consists of 32x32 pixels, three channel images. The images are labeled into ten classes, containing 50,000 training and 10,000 validation images. We implement the network W with four convolutional layers whose input is concatenation of three channel data and four channel watermark (so total of seven channel) and output is a four channeled watermark. D is composed of three transposed convolutional layers, and V has five convolutional layers. For the network of our interest F, F', we use three popular CNN architectures: AlexNet, VGG16, and ResNet50.

Results. We summarize the results in Table I. We observe the followings:

Network Arch.	CIFAR10	Watermarked CIFAR10				
	Baseline	Blend	S	O	D	DeepStamp
AlexNet	82.74	0.5 1.0	78.50 73.51	79.10 73.15	80.13 73.62	79.59 74.09
VGG16	94.00	0.5 1.0	92.71 92.57	92.92 92.83	92.58 92.61	92.74 -
ResNet50	95.37	0.5 1.0	94.88 94.67	94.71 93.64	94.92 93.66	94.18 -

(**S**, **O**, and **D** indicate the **S**tatic method, **O**pacity variation, and **D**isplacement in [2].)

TABLE I. Accuracy of Models Trained with W'marked Data.

- Visible watermarking causes the accuracy drops in all cases, however, if the network capacity (the number of parameters in a network) is higher, the acc. drop is lower.
- When we use a strong blending factor (1.0), the accuracy drop increases. However, the drops are not significant with a high-capacity network (ResNet50).
- Using AlexNet for F, F', our data with synthesized watermark (DeepStamp) has a less accuracy drop than the statically watermarked data (S). However, we are not better when the watermarked data with displacements (D) is used.
- By training VGG16 or ResNet50 (F') with our data synthesized with AlexNet (F), we observe accuracy drops by 1.17% and 0.74% compared to the static (S) method. Since the drops are similar to the case in which we use the same networks as F, once synthesized, the data can be used to train multiple F's.

V. Conclusion

We propose a watermarking framework, DeepStamp, that embeds a visible watermark into the images of interest with less accuracy drop and difficulty of removal for easy claim of the ownership of the watermarked images in the data-sharing scenario. In experiments with the CIFAR10 dataset, we show that the DeepStamp learns transformations of a watermark to be embedded in another images with negligible accuracy drop while making its removal from the images non-trivial.

Acknowledgment

This research is partially supported by Department of Defense and the "Global University Project" grant funded by the Gwangju Institute of Technology (GIST) in 2018.

References

[1] G. W. Braudaway, K. A. Magerlein, and F. C. Mintzer, "Protecting publicly available images with a visible image watermark," in *Optical Security and Counterfeit Deterrence Techniques*, R. L. van Renesse, Ed., vol. 2659, Mar. 1996, pp. 126–133.

[2] T. Dekel, M. Rubinstein, C. Liu, and W. T. Freeman, "On the effectiveness of visible watermarks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2146–2154.

[3] C. Gentry and D. Boneh, *A fully homomorphic encryption scheme*. Stanford University Stanford, 2009, vol. 20, no. 09.

[4] F. Y. Shih, *Digital watermarking and steganography: fundamentals and techniques*. CRC press, 2017.

[5] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 1310–1321.

On the Feasibility of Training Neural Networks with Visibly Watermarked Dataset

Sanghyun Hong*, Tae-hoon Kim^, Tudor Dumitras*, Jonghyun Choi°

*University of Maryland College Park

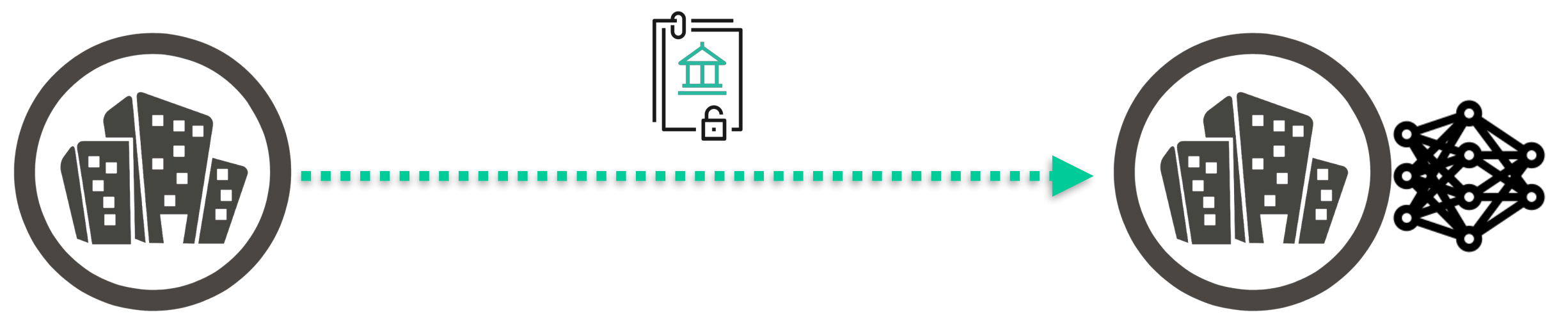
°Gwangju Institute of Science and Technology (GIST)

^Deeping Source Inc.

Overview

- Data collected separately by enterprises and users is hard to be shared with others because *shared data can be stolen*.

Threat Model



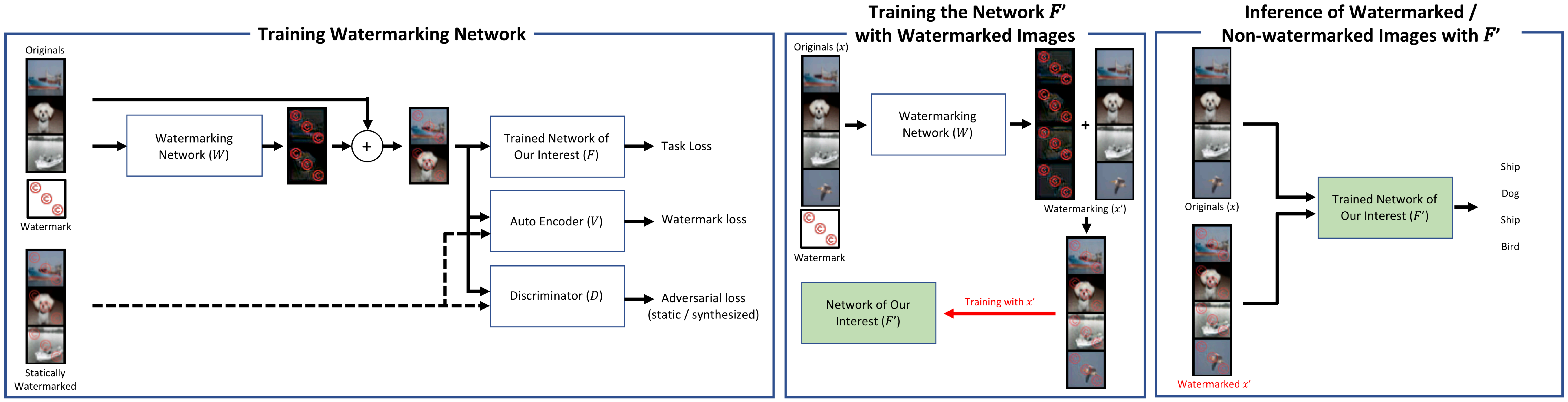
Company A **Company/User B**

- Company A wants to share a dataset collected from the users with Company B
- Company A **still wants to claim the ownership of the data** to prevent Company B resells the data to others or leaks the data

Our Solution

- **Prior work:**
  - Cryptographic solutions (Homomorphic encryption or MPCs) are computationally expensive and, once data is shared, they cannot prevent claiming ownership from B
  - Solutions conceal secrets (Stenography and Invisible watermarking) are susceptible to data augmentations/changes
- **Our solution:**
  - Use *visible watermarking* on datasets shared with other companies and users

DeepStamp Framework



Training Watermarking Network
Originals
Watermark
Statically Watermarked
Watermarking Network (W)
+
Trained Network of Our Interest (F)
Task Loss
Auto Encoder (V)
Watermark loss
Discriminator (D)
Adversarial loss (static / synthesized)
Training the Network F' with Watermarked Images
Originals (x)
Watermark
Watermarking Network (W)
+
Watermarking (x')
Network of Our Interest (F')
Training with x'
Inference of Watermarked / Non-watermarked Images with F'
Originals (x)
Watermarked x'
Trained Network of Our Interest (F')
Ship
Dog
Ship
Bird

Objectives

- Purpose of the visible watermarks on datasets
  - *Visibly Intact:* watermarks embedded to data clearly visible to humans
  - *Hard to Remove:* watermarks are hard to be removed by adversaries
  - *Minimize Accuracy Drops:* a network trained with watermarked data minimizes the accuracy drop compared to the clean network

DeepStamp

- **In Training:**
  - Training a watermarking network (*W*) using clean data and a watermark as inputs with three discriminator networks (*F, V, D*)
  - Discriminators use the data with synthesized watermarks from *W* and the static watermarked data as their counterparts
  - Leverages the Generative Multi-Adversarial Network (GMAN)
- **In Stamping:**
  - Using the trained watermarked network (*W*), synthesize the optimal watermarks for clean data and watermark

Preliminary Results

Network	Baseline	Static	DeepStamp
AlexNet	82.74	78.50	**79.59**
VGG16	94.00	92.71	**92.74**
ResNet50	95.37	**94.88**	94.18

- Using visible watermarking **causes the acc. drops in all cases**, but the drops are minimal when the network capacity is high
- DeepStamp framework can **minimize the accuracy drop further** than the static (additive) watermarking method(s).
- [Results are mentioned in the paper] Our watermarked dataset can be used to train other networks (the data is **transferrable**)