Watermarking-based Defense against Adversarial Attacks on Deep Neural Networks

Xiaoting Li[1], Lingwei Chen[1], Jinquan Zhang[1], James Larus[2], Dinghao Wu[1]
[1]Pennsylvania State University, USA [2]EPFL, Switzerland

***Abstract*—The vulnerability of deep neural networks to adversarial attacks has posed significant threats to real-world applications, especially security-critical ones. Given a well-trained model, slight modifications to the input samples can cause drastic changes in the predictions of the model. Many methods have been proposed to mitigate the issue. However, the majority of these defenses have proven to fail to resist all the adversarial attacks. This is mainly because the knowledge advantage of the attacker can help to either easily customize the information of the target model or create a surrogate model as a substitute to successfully construct the corresponding adversarial examples. In this paper, we propose a new defense mechanism that creates a knowledge gap between attackers and defenders by imposing a designed watermarking system into standard deep neural networks. The embedded watermark is data-independent and non-reproducible to an attacker, which improves randomization and security of the defense model without compromising performance on clean data, and thus yields knowledge disadvantage to prevent an attacker from crafting effective adversarial examples targeting the defensive model. We evaluate the performance of our watermarking defense using a wide range of watermarking algorithms against four state-of-the-art attacks on different datasets, and the experimental results validate its effectiveness.**

***Index Terms*—watermarking, deep neural networks, adversarial examples, defense**

I. Introduction

Deep Neural Networks (DNNs) have been widely adopted in a variety of machine-learning tasks, ranging from computer vision, speech recognition [20], [23] to natural language processing and healthcare [1], [8]. Despite the remarkable performance these applications have achieved, DNNs remain vulnerable to adversarial attacks that design special imperceptible perturbations to the original inputs to fool state-of-the-art models. For example, Goodfellow et al. [17] demonstrated how to add a small perturbation to an image of panda that causes it to be recognized as a gibbon with high confidence. In a security-critical scenario, Evtimov et al. [14] successfully misled a classifier to misclassify a stop sign with some physical perturbations, which can be either graffiti or black and white strips, as a Speed Limit 45 sign.

In order to alleviate adversarial attacks, researchers have proposed a large body of defensive work. Some of them try to manipulate model properties through augmentation or regularization [9], [17], [21], or attempt to filter malicious examples by detecting or removing perturbations introduced to original examples [24], [39]. Most of these strategies are easy to compromise due to their simplicity and differentiable nature, with some impractical assumptions on the attacker's knowledge of the target model. In fact, the information about the target model is the key for most attack algorithms to craft adversarial examples, especially for those gradient-based attacks that require this information to calculate gradients through backpropagation. Recent studies [12], [38] have shown that randomization over the network layerwise structure or inputs enjoy the potential of obfuscating the gradients and thus mitigate the adversarial vulnerability. This naturally inspires us to take advantage of the randomization paradigm and increase the attacker's uncertainty to the target model to significantly hinder them from customizing the model information, such that the generated adversarial examples could be rendered as less effective as possible.

Based on the above observation, in this paper, we consider the practical scenario about the adversarial attacks, and design a defense mechanism by introducing the randomness and confidentiality of digital watermark to DNN models to incur the possible knowledge gap between the attacker and the defender. Digital watermarking is a technique that embeds watermark information into the host image by modifying visually non-significant pixels, which is transparent, imperceptible, and robust. For the watermarking techniques, if a user has no embedding information, the watermark is very challenging to be detected and extracted [31]. In this respect, the attacker needs to craft adversarial examples from their self-trained surrogate models as it is not realistic for them to reproduce the defense model without confidential embedded information. The lack of knowledge about the defense system leads to the discrepancy and stochasticity between the surrogate and real models, making it more challenging for the attacker to successfully evade the defense model. Our proposed defense method enables us to train a DNN model that would not only preserve the inference performance on regular data, but also benefit from knowledge gap and randomization imposed on the learned protocol for better robustness against adversarial attacks. In summary, our work has the following major merits:

- We creatively leverage the concept of knowledge gap by introducing a watermarking system into the DNN model to obstruct the adversarial attacker from accessing the model gradient information. To the best of our knowledge, this is the first investigation to use watermarking techniques to counter adversarial attacks.
- The proposed watermarking-based defense improves the robustness of learning model against adversarial attacks while not compromising its performance on regular data. It is convenient for implementation without many additional

computations and extra training or tuning requirements, and applicable to serve as a general defensive system for different learning models and networks.
- We systematically evaluate our method against adversarial attack algorithms in different scenarios and analyze the impacts of digital watermark on adversarial perturbations. We show that our proposed defense can effectively resist adversarial examples, especially for sophisticated ones.

II. Preliminaries

A. Deep Neural Networks

A deep neural network (DNN) is a function $y = f(x)$ that accepts an input $x \in \mathbb{R}^n$ and produces an output $y \in \mathbb{R}^m$, where f significantly relies on model parameters θ. In our notation, we define f to be the multi-layer neural network structure with the softmax function, which can be denoted as

$$f(x) = \text{softmax}(\sigma_n(W_n\sigma_{n-1}(...\sigma_1(W_1x))) \tag{1}$$

At each layer i, W_i corresponds to model parameters and σ_i is an activation function, usually non-linear, with $1 \leq i \leq n$. In our experiments, we focus on networks that use a ReLU [25] activation function, as it is the most widely used one.

B. Adversarial Examples

Given a valid input x, it is possible to find a similar input x' such that $f(x') \neq f(x)$ yet x and x' are close according to specific distance metric. As such, various adversarial attack methods have been proposed. Fast Gradient Sign Method (FGSM) is designed to fast craft adversarial examples [17]. An example of FGSM attack with respect to a source input x and true label y is

$$x' = x + \epsilon \cdot \text{sign}(\bigtriangledown_x l(x, y)) \tag{2}$$

where $l(x, y)$ is the loss function used to train the classifier, and $\epsilon > 0$ is a small constant that governs the magnitude of distortions. For each pixel in the image, it will take one step of size ϵ in the direction of gradient sign. Projected Gradient Descent algorithm (PGD) is a successful extension of FGSM, which iteratively applies the small FGSM update, with the result being clipped by a sufficiently small constant. Specifically, it begins by setting $x^0 = x$, and then on each iteration k, x^k is updated as

$$x^k = x^{k-1} + \epsilon \cdot \text{sign}(\bigtriangledown_{x^{k-1}} l(x^{k-1}, y)) \tag{3}$$

where $k = 1, ..., K$, and $x' = x^K$. The number of iterations K is determined such that $f(x') \neq f(x)$. DeepFool [27] is another state-of-the-art adversarial example generation approach that projects input x onto the nearest class boundaries iteratively to minimize the Euclidean distance between the input and adversarial examples. In addition, as it has been shown to be very effective in other works, the CW-L2 attack method proposed by [6] is an optimization-based attack that uses L_2-penalty term as its distance metric to find a minimum distortion δ for a given input:

$$\begin{aligned} &\min_{\delta}[\| \delta \|^2 + \lambda_c f(x + \delta)] \\ &s.t. \quad 0 \leq x_i + \delta_i \leq 1 \quad \forall i = 1, ..., N \end{aligned} \tag{4}$$

λ_c is a suitable constant chosen by binary search, $x + \delta$ represents the adversarial example x' we would like to find, and $f(\cdot)$ is an effective objective function

$$f(x') = \max(-\kappa, \max\{Z(x')_{f(x)} : t \neq f(x)\} - Z(x')_t) \tag{5}$$

where κ denotes a margin parameter that controls the confidence in result, and $Z(x)_t$ is the logit (the value before the softmax layer) corresponding to class t.

C. Digital Watermark

Digital watermarking is a technique used for the protection of digital work such as video, audio, and image [35]. In this technique, a secret payload (i.e., watermark) is embedded to the work using some watermarking algorithm that should be imperceptible, robust, and of high fidelity. Specifically, a watermarking algorithm consists of a watermark structure and an embedding algorithm. According to the modified value of the carrier, digital watermarking is divided into two major areas: spatial domain watermarking and frequency domain watermarking. For the sake of the imperceptibility and robustness, current image watermarking research mainly focuses on frequency domain watermarking techniques, where the image is represented as the form of frequency, and the watermark is embedded into the coefficients of the transformed image. In general, the frequency domain transform is considered to be more robust than that in spatial domain. Therefore, we explore a couple of frequency domain watermarking transforms in our defensive strategy.

III. Proposed Method

In this section, we present the detailed approach of how to design a watermarking-based defense and how to enhance the target model's robustness against adversarial attacks based on such strategy.

A. Knowledge Gap

Given a DNN model f, the output label y for an input x is presented by $y = \arg\max_i f_i(x)$, where $f_i(x)$ is the confidence score of the predicted label i. Any adversarial attack that aims to alter the output label y of the model regarding the input sample x needs to change the confidence score $f_y(x)$ by adding the perturbation δ to x, such that the output prediction is changed by a fixed lower bound ε: $\|f_y(x) - f_y(x+\delta)\|_2 \geq \varepsilon$. According to the first-order approximation [10] and the DNN model's linear characteristics around the input samples [15], the difference caused by perturbation δ on $f_y(x)$ can be denoted as $f_y(x + \delta) - f_y(x) \approx \langle \nabla_x f_y, \delta \rangle$. Therefore, the minimal l_p-norm perturbation $\hat{\delta}_p$ ($p \in [1, \infty)$) required to change the output prediction by ε can be approximated using Hölder inequality and l_p-norm projection as [10]:

$$\hat{\delta}_p \approx (\frac{\varepsilon}{\|\nabla_x f_y\|_q})\partial(\|\nabla_x f_y\|_q) \tag{6}$$

where l_p-norm and l_q-norm are dual-norm with $\frac{1}{p} + \frac{1}{q} = 1$, and $\partial(\cdot)$ is the subgradient of the argument. Dabouei et al. [10]

provided us with a more detailed solution of $\partial(\|\nabla_x f_y\|_q)$, such that Eq. (6) can be rewritten as [10]:

$$\hat{\delta}_p \approx (\frac{\varepsilon}{\|\nabla_x f_y\|_q})(\frac{|\nabla_x f_y|^{q-1} \odot \text{sign}(\nabla_x f_y)}{\|\nabla_x f_y\|_q^{q-1}}) \quad (7)$$

Clearly, we can gain the insight from Eq. (7) that the model gradient $\nabla_x f_y$ plays a very important role in the success of the adversarial attacks; it is essential to obfuscate $\nabla_x f_y$ to defend against such attacks. With this in mind, some significant efforts have been made in this regard to enforce useless gradients for generating adversarial examples [2], while randomized defenses [12], [38] with randomization over the network structure or inputs can restrain the attacker from correctly estimating the true gradient and thus failing to effectively mislead the model.

More specifically, due to the stochastic gradient caused by randomized model structure or inputs, the attacker cannot directly customize the defense model $f(x)$ but train their own model $\hat{f}(\hat{x})$ to compute the gradient $\nabla_{\hat{x}} \hat{f}_{\hat{y}}$. As investigated, adversarial examples may be transferable so that some adversarial examples generated from $\hat{f}(\hat{x})$ may lead to misclassification on $f(x)$ as well [30]. Such a property allows the attacker to take $\hat{f}(\hat{x})$ as a surrogate model to craft attack samples. However, since the surrogate model is a rough approximation of the target distribution (i.e., $\nabla_{\hat{x}} \hat{f}_{\hat{y}} \neq \nabla_x f_y$), there is always a discrepancy between the approximation and the real one, which we consider as the defense space. Our goal is to enlarge such space, deviate $\nabla_{\hat{x}} \hat{f}_{\hat{y}}$ from $\nabla_x f_y$, and make the adversarial perturbation less effective. Different from the previous studies, here we devise a fine-grained watermarking system to the DNN model to increase the knowledge gap between the attacker and the defender. As the watermark is transparent and undetectable with secret payload message and capacity, the embedded watermark information may cause the inputs x and the corresponding gradient $\nabla_x f_y$ randomized, and expand the discrepancy between $f(x)$ and $\hat{f}(\hat{x})$. The DNN model $f(x)$ could thus explicitly change its classification boundary, and be resilient against the attacker's adversarial examples generated through the untrue estimation $\hat{f}(\hat{x})$.

B. Watermarking-based Defense

For defenses that employ randomized transformations to the inputs, Athalye et al. [2] demonstrated that Expectation over Transformation (EOT) can be deployed to compute the gradient over the expected transformation to the inputs by optimizing the expectation over the transformation $\mathbb{E}_{t\sim T} f(t(x))$ (i.e., $t(\cdot)$ sampled from a distribution of transformations T). However, the distribution T can merely model perceptual transformations, such as image cropping, viewpoint shifts and geometric changes. Different from the regular input transformations, watermark embedding, which imposes the random and secret payload message to the inputs in the abstract frequency domain, is imperceptible and irreversible for the attacker, and its distribution T is thus difficult to be formulated. Even if some of the watermarked data is accidentally intercepted by the attacker, they are still unable to detect

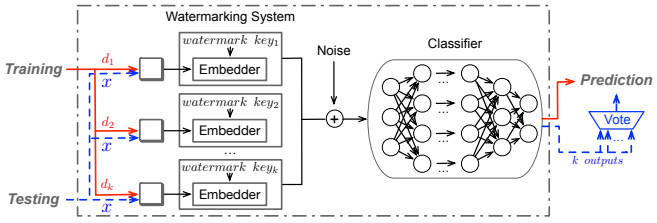

Fig. 1: The overview of our defense framework devising a watermark system between the input and the DNN structure.

and extract all the watermark keys for reproduction. In this respect, the gradient obfuscation and knowledge gap caused by watermarking are effective to prevent adversarial attacks from directly or indirectly calculating model gradients.

In order to generate additional randomization benefits, instead of applying watermarking to the inputs (either clean or adversarial) using an unique payload message, we design our watermarking defense paradigm in an ensemble manner, the overview of which is illustrated in Figure 1. (1) In the training stage, we randomly split the training data into k sets; for each set, we embed a watermark key into all inputs; different sets use different keys, while all inputs enjoy the same embedder (a watermarking algorithm). All watermarked sets are leveraged for DNN model training. (2) Complying to the watermarking routine, we add small Gaussian noise with the interval of $[-0.1, 0.1]$ to each watermarked input and then rescale it to increase the difficulty of watermark detection. (3) In the testing stage, we feed the test data through watermarking system using k specified watermark keys, and the trained DNN model to obtain k different outputs respectively, which are aggregated later using voting method to approximate the final result. This may generate some regularization effect beyond randomization provided by model training. Algorithm 1 illustrates our proposed watermarking-based defense.

It's worth noting that the embedded secret watermark has no significant impact on the performance of the DNN model on regular classification task in our observation. In addition, the watermark keys are images randomly selected from the large image database (e.g., ImageNet [11]), which is independent from the input data. We further resize the watermark images into the same shape as the input data before embedding them. Following the second Kerckhoffs' cryptographic principle [26], we err on the side of overestimating the attacker's capability and excessively relax the limitation on the attacker's knowledge about the defense model (i.e., the worst-case where the watermarking algorithm and the watermark image database are also known to the attacker). As such, the attacker tends to search for the potential watermark keys to craft effective adversarial examples. The computational space for watermark searching would be as large as $k \times 256^{h\times w\times d}$, where 256 is the range of image pixel and $[h, w, d]$ represents the image shape. It forces the attacker to take an extremely long time and effort to evade the target model. Therefore, it is computationally infeasible to generate adversarial examples in such a cost-expensive fashion. Unlike the low-level image transformations [38], the watermarking also implicitly preserves

Algorithm 1: Watermarking-based defense.

Input: D_{tr}: training data, D_{ts}: test data (clean or adversarial), f: a standard DNN model, $\{w\}_{i=1}^{k}$: k random watermark images, g: a watermarking algorithm, μ: Gaussian noise.

Output: $\widehat{f}$: a defense model; y: output class.

$\{w\}_{i=1}^{k} \leftarrow$ convert $\{w\}_{i=1}^{k}$ to gray-scale images;
//For the training stage;
$\{D_{tr}\}_{i=1}^{k} \leftarrow$ split D_{tr} into k sets;
for $i = 1 \rightarrow k$ **do**
  for $j = 1 \rightarrow |D_{tr}^{i}|$ **do**
    $\widehat{x}_j = g(x_j, w_j) + \mu$;
  end
end
Train DNN model f using the watermarked $\widehat{D}_{tr}$ as $\widehat{f}$;
//For the testing stage;
for $i = 1 \rightarrow |D_{ts}|$ **do**
  for $j = 1 \rightarrow k$ **do**
    $\widehat{x}_{ij} = g(x_i, w_j) + \mu$;
    $y_{ij} = \widehat{f}(\widehat{x}_{ij})$;
  end
  $y_i \leftarrow$ aggregation using voting on $\{y_{ij}\}_{j=1}^{k}$;
end
return The trained model $\widehat{f}$ and the test classes y;

the specific structure and meaningful pattern, which codes better with the uniqueness of the input images in our defense model and generates a better advantage to avoid the attacker's mimicry.

C. Watermarking Implementation

When embedding a watermark to different images, we have to adopt a specific watermarking strategy for implementation. In this work, we investigate five different frequent domain watermarking algorithms in our watermarking system as follows.

- Discrete Fourier Transform (DFT) [36] decomposes an image in sine and cosine form. Since the magnitude and phase hold some information of the transformed image, we can accordingly modify them to embed the watermark.
- Discrete Wavelet Transform (DWT) [32] gives a multi-resolution representation of the image, which divides the image into high-frequency quadrant and low-frequency quadrant. We embed the watermark into low-frequency coefficients as they contain the details of the original images.
- Singular Value Decomposition (SVD) [33]. Given an image matrix, it can be transformed into three components, where the most significant coefficients in the component are modified to embed a watermark. Afterwards, it is inversely transformed to reconstruct the watermarked image.
- DWT_SVD algorithm [16] develops the DWT and SVD methods, which is a technique that clubs the properties of DWT and SVD. It not only increases the limited capacity of SVD but also reduces time consumption.
- DWT_DCT_SVD algorithm [28] combines DWT, DCT and SVD and is robust against geometric attacks.

Besides the watermarking algorithms, the implementation of watermarking also significantly relies on the property of the data, e.g., the dimension of the image. The gray-scale image watermarking is convenient for implementation since all the intrinsic information of the gray-scale images is simply abstracted as pixels in a single component. Differently, the color image are generally represented as a red-green-blue (RGB) triplet, while the RGB values are more complex and are the only feasible data from them [7]. Considering that these three components are inter-correlated and RGB triple is also a biased representation of the color images, processing the RGB color information in parallel for each color component independently while ignoring the intrinsic properties contained in the interaction of different color channels may easily enforce information loss and thus lead to model performance degradation. To address this issue, we attempt two mapping solutions to transfer the color information into independent components instead of the $R-G-B$ components.

The first one is inspired by the work [3], where we employ Karhunen–Loeve Transform (KLT) to decorrelate RGB information of the color images. To apply KLT, each image is represented as a set of vectors $\mathbf{v}_i$ of size d (e.g., $d = 32 \times 32$ if the dimension of the color image is 32×32), with $1 \leq i \leq 3$. As such, it is possible to calculate the expected value of three vectors as follows:

$$\mathbf{m} = E[\mathbf{v}_i], \tag{8}$$

which would further facilitate computing the covariance matrix of size 3×3 of the centered vectors $(\mathbf{v}_i - \mathbf{m})$:

$$\mathbf{C} = E[(\mathbf{v}_i - \mathbf{m}) \cdot (\mathbf{v}_i - \mathbf{m})^T], \tag{9}$$

where the eigenvectors $\mathbf{a}_i$ and their associated eigenvalues λ_i of the matrix $\mathbf{C}$ can be obtained to formulate a matrix $\mathbf{A}$ by descending order of the eigenvalues as $\mathbf{A} = ({\mathbf{a}_1}^T, {\mathbf{a}_2}^T, {\mathbf{a}_3}^T)$. The KLT of a vector $\mathbf{v}_i$ can be defined as

$$\mathbf{u}_i = \mathbf{A} \cdot (\mathbf{v}_i - \mathbf{m}), \tag{10}$$

and these vectors are uncorrelated [3]. Here we embed the watermark into the first component $\mathbf{u}_1$ after KLT transformation as it generally contains the most information, and then an inverse KLT is performed to reconstruct the watermarked images in the way:

$$\mathbf{v}_i = \mathbf{A}^{-1} \cdot \mathbf{u}_i + \mathbf{m}. \tag{11}$$

The second solution is mapping correlated RGB components to HSV [4], a color space designed to more closely align with the way human vision perceives color. HSV describes colors in terms of Hue, Saturation, and Value. Considering that HSV is a less correlated color space than RGB while objects in images have distinct hues and luminosities so that these features can be used to separate different image areas, we choose to convert RGB triple to HSV, embed the watermark to the Hue value, and then map HSV components back to RGB to construct the watermarked images.

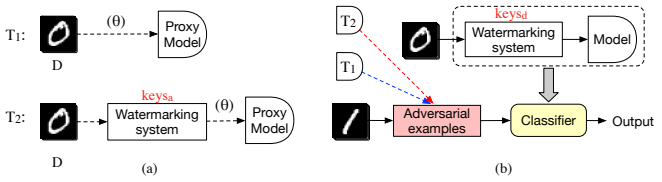

Fig. 2: (a): Two threat models (zero knowledge threat model T_1 and partial knowledge threat model T_2); (b): the evaluation workflow where the defense model is trained on watermarked data and different attack models generate adversarial examples to attack the defender.

IV. Evaluation

In this section, we evaluate the efficacy of our proposed watermarking based defense model through performing experiments on benchmark image classification tasks, and compare our model with a wide variety of state-of-the-art approaches.

A. Experimental Setup

Datasets. We test our model on three benchmark image classification datasets from the AI Science community: MNIST [22], Fashion-MNIST [37] and CIFAR-10 [19]. MNIST is a set of hand-written digits that contains 10 classes, 60,000 training and 10,000 test gray-scale images of the size 28×28, while Fashion-MNIST [37] is another standard dataset with more complex and diverse object structures, which also consists of 10 classes, 60,000 training and 10,000 test gray-scale images of 28×28. CIFAR-10 [19] is composed of 60,000 32×32 colour images in 10 classes. To be consistent with the previous work, we scale each pixel value to be in the range $[0, 1]$.

DNN model and watermarking. The DNN model trained in our experiments is a standard CNN classifier with 8 layers [29]. The architecture of our deployed model is not complicated because our ultimate goal is not to achieve the state-of-the-art image classification accuracy on the chosen dataset but measure the performance of classifying adversarial examples in the same settings. The overall classifier performance is calculated by the test accuracy. We evaluate our defensive method on four state-of-the-art adversarial attacks, i.e., FGSM [17], PGD [21], DeepFool [27] and CW-L2 [5]. We set the number of watermarking embedders k in our system as 5. The watermark images used in either the attack model or the defense model are randomly selected from the ImageNet [11] with more than 14 million images, and processed through the watermarking system upon the dataset. To verify if watermarked training has any impact on the learning performance, we test the classification accuracy of our trained DNN system on the regular image data. We found that the classification results of different training dataset over different watermarking algorithms are consistent with the benchmark in standard case, where the test accuracy reaches 99.30% over MNIST, 91.49% over Fashion-MINIST and 88.30% over CIFAR-10 on average.

Attack ability. In our work, we assume the attacker could obtain most of the essential information about the target model, such as DNN structure, training raw data D, hyperparameters θ used for model training, and the classification output, but is incapable of probing the internal variables of the network to gain access to the watermark image or the watermarked input in the continuous workflow. This assumption is reasonable taking account of the modern protected computing systems and it is typical in domains, for example, biometric and digital watermarking applications. We evaluate our defense strategy against the attackers with different amounts of knowledge about our defense method. As watermarking color images is a challenge compared to gray-scale images, we place color watermarking defense in a separate experimental section.

B. Evaluation of Watermarking-based Defense

Defense against Zero Knowledge Attack T_1. In the first scenario, we consider a very straightforward attack type T_1 (Figure $2(a)$), where the attacker has zero knowledge about defense. Intuitively, the attacker utilizes the obtained training set D and hyperparameters θ from the observation, and trains a surrogate DNN model within their knowledge for the same image recognition task as the target model does. The whole evaluation process is illustrated in Figure $2(b)$, where the attacker generates adversarial examples from the trained threat model T_1 to attack the defense classifier, while the classifier contains a watermarking system and a well-trained DNN model, i.e., the adversarial examples will be first processed with the watermarking system before fed for classification. In this part of experiments, we compare the results of our method encoded with five different watermarking embedders including DFT, DWT, SVD, DWT_SVD and DWT_DCT_SVD with a standard DNN classifier without watermarking defense, trained on the original input image set, whose results also serve as the baseline. We craft 1,000 adversarial examples, and compute the test accuracy of our defense model on these generated adversarial examples. The results are shown in Table I. We can observe that:

- DFT can effectively decrease the classification error of adversarial examples for all types of considered attacks. Specifically, it enhances the model's test accuracy on adversarial examples from 0.6–4.8% to 48.3–92.3% on MNIST. In the case of Fashion-MNIST, the results have slightly declined, but we can still see an improvement against a variety of adversarial examples (35.4–75.1% increase). For FGSM and PGD, DFT-based defense significantly outperforms other methods.
- Our method obtains outstanding results on DeepFool and CW-L2 for all employed watermarking algorithms. On MNIST, the test accuracy increases up to 95.6% on DeepFool and 93.3% on CW-L2. On Fashion-MNIST, the best defense result reaches 84.3% test accuracy on DeepFool and 82.0% on CW-L2. By contrast, the experimental results on FGSM and PDG are not as good as that on DeepFool and CW-L2.

Defense against Partial Knowledge Attack T_2. In the second attack scenario, we focus on a stronger attack, where we enable the attacker to partially learn about our defensive strategy

TABLE I: Classification accuracy (%) of defense model against attack T_1

Watermarking	MNIST				Fashion-MNIST			
	FGSM	PGD	DeepFool	CW-L2	FGSM	PGD	DeepFool	CW-L2
–	4.8	0.6	1.0	0.6	6.4	5.2	6.8	6.3
DFT	**60.7**	**48.3**	94.5	92.3	**47.9**	**40.6**	79.5	81.4
DWT	58.9	28.5	**95.6**	**93.3**	21.7	14.4	82.6	81.9
SVD	56.6	13.7	94.7	82.2	24.5	18.8	80.0	74.7
DWT_SVD	53.2	26.3	93.6	90.4	28.7	26.6	82.0	81.9
DWT_DCT_SVD	49.3	23.6	94.1	91.9	37.0	37.4	**84.3**	**82.0**

TABLE II: Classification accuracy (%) of defense model against attack T_2

Watermarking	MNIST				Fashion-MNIST			
	FGSM	PGD	DeepFool	CW-L2	FGSM	PGD	DeepFool	CW-L2
–	4.8	0.6	1.0	0.6	6.4	5.2	6.8	6.3
DFT	**62.7**	**44.2**	90.1	84.6	**48.3**	**39.3**	78.3	71.5
DWT	51.9	29.5	91.7	80.9	19.7	12.8	80.4	70.9
SVD	42.7	13.3	94.6	86.8	22.9	16.8	81.2	75.2
DWT_SVD	49.4	22.9	92.1	88.9	26.2	23.5	78.8	70.1
DWT_DCT_SVD	43.1	27.6	**95.4**	**93.4**	34.0	36.9	**83.4**	**76.4**

including watermarking embedder devised in the defense method, but not the watermark keys embed to the input data. We define the second threat model as T_2 (as presented in Figure 2(a)). Specifically, to substitute the watermark images $keys_d$ used in defense model, the attacker randomly picks surrogate images $keys_a$ from the large image database to watermark the input, and trains their surrogate model for adversarial example generation. The evaluation process for our defense model against threat model T_2 is depicted in Figure 2 as well, where the attacker trains the network on the watermarked image set embedded with a different watermark set $keys_a$ from $keys_d$ through the same watermarking system as the defense framework. Likewise, five watermarking algorithms are performed and compared on 1,000 adversarial examples for evaluation. The experimental results are shown in Table II, where we can see that:

- DFT can potentially improve the robustness of the model against FGSM and PGD better than other watermarking algorithms. The results show a 43.6–57.9% accuracy increase on MNIST. For more complex Fashion-MNIST, it shares the same observed tendency, but with a slight drop-off.
- Similar to Attack I, our defense achieves very promising results against DeepFool and CW-L2 for all the watermarking algorithms. On MNIST, it can decrease the classification error on adversarial examples from 99.0–99.4% to 4.6–19.1%. For Fashion-MNIST, the error rate is reduced from 93.2% to 16.6% on DeepFool and from 93.7% to 23.6% on CW-L2.

Discussion. The experimental results and analysis demonstrate that watermarking-based defense can effectively enhance DNN robustness against adversarial attacks, even the attacker may have different knowledge about the targeted system. In particular, our method achieves high performance against the sophisticated attacks, e.g., DeepFool and CW-L2. Unlike other attacks, these attacks are optimally generated through iterative optimization, which may easily get overfitted to the model parameters and training dataset and result in weak generalization. As a result, the delicate changes brought by the watermark have a significant impact on them.

On the other hand, the defense efficacy on FGSM and PGD attacks underperform DeepFool and CW-L2, since such attacks have lower variance and better transferability to the learning models. Also, the information discrepancy caused by the subtle watermark does not necessarily induce sufficient patterns to destroy the specific structure of adversarial perturbations. To address this limitation, we might need to either introduce more potentially secret information to enlarge the knowledge gap between the defender and the attacker, or leverage additional techniques (e.g., adversarial training against single-step attacks) to further facilitate our watermarking-based defense. We leave it as our future work.

C. Watermarking Defense on Color Images

Watermarking RGB images is not as straightforward as that on gray-scale images, due to the fact that these three components are inter-correlated. To put it into perspective, we initially exploit DFT to watermark RGB component for the color images from CIFAR-10 [19], train a DNN model with and without color watermark embedding process on regular data. The test accuracy decreases from 88.3% to 38.1%. Such a naive watermarking method almost generates a denial of service for classification, let alone be used as defense against adversarial attacks. In our work, we attempt KLT and HSV transform to decorrelate RGB information of the color images in our defense, respectively. We preliminarily assess the effectiveness of these color image watermarking methods on the adversarial examples generated by different adversarial attacks on CIFAR-10. The results are showed in Figure 3.

From Figure 3, we can see that the embedding process using transformations indeed helps to improve the color watermarking quality, which outperforms the direct watermarking on

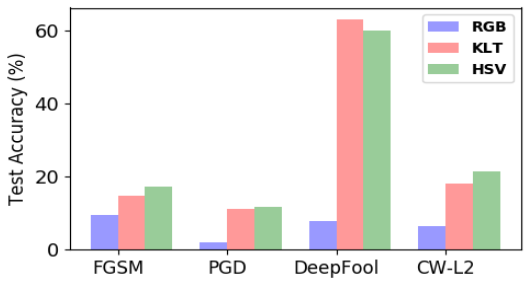

Fig. 3: Accuracy on different color transformations.

RGB space. Regarding the adversarial examples, our defense using KLT and HSV improves the classification performance by different degrees against different attacks, especially that the test accuracy achieves 63.2% and 60.0% against DeepFool, which is better than other attacks. The same observations can be found in Table I and Table II, and the defense efficacy difference among FGSM, PGD, DeepFool and CW-L2 has been well explained in Section IV-B. Considering that color image watermarking is still an open issue with challenges and difficulties (e.g., color representations) [7], we'd like to gain further insight to enhance the color watermarking in our defense strategy in the future work.

D. Comparisons with Other Methods

In this set of experiments, we examine the effectiveness of our defense model against the adversarial attacks by comparisons with other related state-of-the-art defense methods, including: (1) resizing [38], (2) padding [38], (3) resizing+padding [38], (4) bit-depth reduction [18], [39], (5) JPEG compression [13], [18], and (6) Guassian noise [34]. More specifically, resizing strategy resizes the original input images into a new image with random size. As discussed in [38], the difference between the original and new sizes should be within a reasonably small range to avoid performance drop-off. Considering the image set used in our experiments is of 28×28 size, we set the new size for each image as 30×30. Padding pads zeros around the resized images for each side. For resizing+padding, we first resize the images to 29×29, and then pads zero pixels on the left and bottom to obtain 30×30 images. Bit-depth reduction performs a type of quantization to squeeze image features that can possibly remove small adversarial perturbations; we reduce the images to 4 bits in our experiments. JPEG compression uses the similar way to disrupt adversarial perturbations; we follow the work [18] to perform compression at quality level 75 (out of 100). Guassian noise $\mathcal{N}(0, 1)$ is added to the image data to introduce randomization to the target model. The experimental results on MNIST using DFT watermarking algorithm are reported in Table IV.

From Table III, we observe that different image transformations can mitigate the adversarial effects for iterative attacks like DeepFool and CW-L2 significantly, the reason behind which has been well analyzed in Section IV-B. As for FGSM and PGD indicating stronger transferability, these alternative

TABLE III: Accuracy (%) over different defense models

Defense Methods	FGSM	PGD	DeepFool	CW-L2
–	4.8	0.6	1.0	0.6
Resizing	14.7	2.0	92.8	92.1
Padding	13.4	0.9	93.4	92.6
Resizing+Padding	14.5	0.7	90.4	91.9
Bit-depth reduction	7.9	0.6	85.0	72.4
JPEG compression	11.1	0.7	93.3	82.9
Guassian Noise	10.5	0.6	92.2	86.0
Watermarking	**60.7**	**48.3**	**94.8**	**93.6**

TABLE IV: Accuracy (%) over different watermark classes

Watermark Class	FGSM	PGD	DeepFool	CW-L2
Tobacco shop	53.7	44.7	89.9	95.3
Tractor	54.0	36.4	91.8	93.7
Pug	52.4	41.4	92.0	94.3
Vase	47.8	27.7	91.6	94.9
Gorilla	49.9	31.6	88.2	95.1
Valley	50.7	36.5	93.3	82.9

image transformation methods suffer from a drastic drop-off, i.e., the best classification accuracy can only reaches to 14.7% and 2.0% for FGSM and PGD respectively. By contrast, our watermarking-based defense can well preserve the unique structure and patterns through the designed watermarking procedure and enforce a distinctive discrepancy between the defense model and the surrogate model, and thus outperforms other related defense methods.

E. Evaluation on Different Watermark Patterns

Due to the large amount of possible patterns introduced by watermarking system, it is worth analyzing the different types of watermark images that work for our defense model precisely. In this section, we thus validate the effectiveness and significance of watermark image patterns in building a defense model. In our experiments, we limit the freedom of watermark image choice to be one class of images from ImageNet, and randomly choose different watermark images of one specific class to be watermark keys. We test the watermarking system encoded with six image patterns respectively (i.e., tobacco shop, tractor, pug, vase, gorilla, valley) to evaluate the performance of the defense model against adversarial attacks. As illustrated in prior experiments, DFT performs better than other four transformation algorithms applied in our defense strategy on average; therefore, we evaluate the effectiveness of different watermark images using DFT as the embedder of watermarking system. We report the results with respect to the classification accuracy on MNIST in Table IV.

As revealed from the results, the defense performances slightly vary in different classes of watermark images where some image patterns could outperform others against one adversarial attack while underperform a bit against another attack (e.g., Tobacco shop achieves 95.3% accuracy on CW-L2 while 89.9% accuracy on DeepFool). Overall, watermarking-based defense is not strictly sensitive to the specific patterns introduced by the watermark images, and is able to reach reasonable performance under a random image choice. Recall that, the watermarking is also easy for implementation with-

out many additional computations and extra training. These properties make our defense model convenient and feasible in practical use.

V. Conclusion and Future Work

In this paper, we propose a watermarking-based defense mechanism against adversarial examples by imposing a secret watermarking system into the DNN model to yield a knowledge gap advantage. The experimental results demonstrate that our defense can effectively enhance the robustness of the DNN classifier against adversarial attacks, and prove that watermark is a good choice to introduce randomization of the defense model. In addition, we show a promising potential of our proposed idea against adversarial attacks from limiting the attacker's knowledge of the defense model, especially for the optimized adversarial perturbations. On the other hand, our defense underperforms on some adversarial examples, such as PGD. As future work, we aim to investigate other potential methods to enlarger the amount of secret information or assemble additional techniques (e.g., adversarial training) to further facilitate the model protection and examine its behavior on more complex datasets.

VI. Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This research was also supported in part by a seed grant from the Penn State Center for Security Research and Education (CSRE).

References

[1] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally normalized transition-based neural networks," *arXiv preprint arXiv:1603.06042*, 2016.

[2] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*. PMLR, 2018.

[3] A. Basso, D. Cavagnino, V. Pomponiu, and A. Vernone, "Blind watermarking of color images using karhunen–loève transform keying," *The Computer Journal*, 2010.

[4] D. Cardani, "Adventures in hsv space," *Laboratorio de Robótica, Instituto Tecnológico Autónomo de México*, 2001.

[5] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017.

[6] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *S&P*. IEEE, 2017, pp. 39–57.

[7] G. Chareyron, J. D. Rugna, and A. Tremeau, "Color in image watermarking," *Advanced Techniques in Multimedia Watermarking: Image, Video and Audio Applications*, 2010.

[8] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[9] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *ICML*, 2017, pp. 854–863.

[10] A. Dabouei, S. Soleymani, F. Taherkhani, J. Dawson, and N. M. Nasrabadi, "Exploiting joint robustness to adversarial perturbations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1122–1131.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009.

[12] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," *arXiv preprint arXiv:1803.01442*, 2018.

[13] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[14] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on deep learning models," *arXiv preprint arXiv:1707.08945*, vol. 1, p. 1, 2017.

[15] A. Fawzi, S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto, "Empirical study of the topology and geometry of deep networks," in *CVPR*, 2018, pp. 3762–3770.

[16] E. Ganic and A. M. Eskicioglu, "Robust DWT-SVD domain image watermarking: embedding data in all frequencies," in *Proceedings of the 2004 Workshop on Multimedia and Security*, 2004, pp. 166–174.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[18] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.

[19] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[21] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[22] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998.

[24] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No need to worry about adversarial examples in object detection in autonomous vehicles," *arXiv preprint arXiv:1707.03501*, 2017.

[25] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, vol. 30, no. 1, 2013.

[26] J. L. Massey, "Cryptography: Fundamentals and applications," in *Copies of transparencies, Advanced Technology Seminars*, vol. 109, 1993.

[27] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *CVPR*, 2016.

[28] K. Navas, M. C. Ajay, M. Lekshmi, T. S. Archana, and M. Sasikumar, "DWT-DCT-SVD based watermarking," in *COMSWARE*, 2008.

[29] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.

[30] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[31] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *INDIN*. IEEE, 2005, pp. 709–716.

[32] M. J. Shensa, "The discrete wavelet transform: wedding the a trous and mallat algorithms," *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2464–2482, 1992.

[33] R. Sun, H. Sun, and T. Yao, "A svd-and quantization based semi-fragile watermarking technique for image authentication," in *Signal Processing, 2002 6th International Conference on*, vol. 2. IEEE, 2002.

[34] O. Taran, S. Rezaeifar, and S. Voloshynovskiy, "Bridging machine learning and cryptography in defence against adversarial attacks," in *European Conference on Computer Vision*. Springer, 2018.

[35] R. G. Van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *ICIP*, vol. 2, 1994, pp. 86–90.

[36] S. Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete fourier transform," *IEEE transactions on Communication Technology*, vol. 19, no. 5, pp. 628–634, 1971.

[37] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[38] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv:1711.01991*, 2017.

[39] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *preprint arXiv:1704.01155*, 2017.