Exploiting vulnerabilities of deep neural networks for privacy protection

Ricardo Sanchez-Matilla, Chau Yi Li, Ali Shahin Shamsabadi, Riccardo Mazzon, Andrea Cavallaro

Abstract—Adversarial perturbations can be added to images to protect their content from unwanted inferences. These perturbations may, however, be ineffective against classifiers that were not seen during the generation of the perturbation, or against defenses based on re-quantization, median filtering or JPEG compression. To address these limitations, we present an adversarial attack that is specifically designed to protect visual content against unseen classifiers and known defenses. We craft perturbations using an iterative process that is based on the Fast Gradient Signed Method and that randomly selects a classifier and a defense, at each iteration. This randomization prevents an undesirable overfitting to a specific classifier or defense. We validate the proposed attack in both targeted and untargeted settings on the private classes of the Places365-Standard dataset. Using ResNet18, ResNet50, AlexNet and DenseNet161 as classifiers, the performance of the proposed attack exceeds that of eleven state-of-the-art attacks. The implementation is available at https://github.com/smartcameras/RP-FGSM/

Index Terms—Deep learning, adversarial images, privacy protection.

I. INTRODUCTION

Images shared online capture people and scenes that reveal personal information, as well as information about personal choices and preferences. This information can be automatically inferred by classifiers. To prevent this potential privacy violation and to protect the visual content from unwanted automatic inferences, we aim to exploit the vulnerability of classifiers to adversarial attacks.

An adversarial attack should mislead classifiers that the attacker has access to (seen classifiers), as well as classifiers that the attacker has no information about, not even the prediction output (unseen classifiers). However, adversarial attacks often fail to mislead unseen classifiers as the generated adversarial perturbations overfit to a specific classifier or to an ensemble of classifiers. Adversarial images should also be robust to defenses. Defenses can be based on re-quantization, median filtering and JPEG compression. Moreover, adversarial perturbations should not degrade the image quality, especially when added for protecting privacy.

In this paper, we propose an adversarial attack that aims to prevent both seen and unseen classifiers from inferring private information present in an image, even when the classifiers are equipped with known defenses. At the core of the proposed attack there is an iterative combination of random selections of a classifier and a defense within a Fast Gradient Signed Method (FGSM) framework. This random selection avoids the creation of perturbations that overfit to a specific classifier or defense, and improves the misleading rate for seen and unseen classifiers. The proposed attack, which can work in both targeted and untargeted settings, is related to methods that are based on ensemble of classifiers and on defense transformations, but differs in the fact that both classifiers and transformations are randomly chosen at each iteration. Moreover, the proposed attack enables the use of complex defense transformations with null derivative. We validate the proposed attack for the protection of scene content on the privacy subset of the Places365-Standard dataset, which consists of scenes such as places of worship and hospitals. We evaluate the attack on state-of-the-art classifiers, namely ResNet18, ResNet50, AlexNet, and DenseNet161, based on the misleading rate, detectability and image quality.

II. PROBLEM DEFINITION

Let \( x \in \mathbb{R}^{W \times H \times C} \) be an image of width \( W \) and height \( H \) pixels, and \( C \) color channels whose dynamic ranges are \([0, 255]\). Let \( M(\cdot) \) be a \( D \)-class deep neural network classifier with parameters \( \theta \) and trained using the cost function \( J_M(\cdot) \).

Let \( \hat{y}_x \) be the true class associated with \( x \). The classifier outputs a prediction vector for \( x \), \( \mathbf{p}_x = (p_i)_{i=1}^D = M(x) \), where \( p_i \) represents the probability of \( x \) being associated with class \( i \in \{1, ..., D\} \). The predicted class for \( x \), \( y_x \), is the most likely of \( D \) classes:

\[
y_x = \arg\max_{i=1, ..., D} p_i. \tag{1}
\]

Note that the predicted class might differ from the true class, which is unknown during the execution of the adversarial attack. Adversarial attacks for privacy protection should aim to hide the true class, \( \hat{y}_x \), from \( M(\cdot) \), even when the predicted class is incorrect. The adversarial perturbation, \( \delta_x \in \mathbb{R}^{W \times H \times C} \), added to the original image, \( x \), generates an adversarial image, as \( \tilde{x} = x + \delta_x \). This perturbation causes the classifier to predict an adversarial class, \( y_x \), by decreasing the probability of the predicted class, \( y_x \), (untargeted attack) until

\[
y_x \neq y_x, \tag{2}
\]

or by increasing the probability of a specific target class, \( y_t \), (targeted attack), until

\[
y_x = y_t \neq y_x. \tag{3}
\]

\(^1\)FGSM is the iterative attack proposed by Kurakin et al. Note that this attack is also known as Basic Iterative Method (BIM) and as Projected Gradient Descent (PGD) attack with \( L_\infty \) norm.
The target class can be determined randomly [1], [12], systematically as the least-likely class [12], or adaptively from the prediction vector [1].

Defenses against adversarial attacks aim to eliminate, prior to inputting images to the classifier, the effect of possible adversarial perturbations using a transformation, \( \phi(\cdot) \), namely median filtering [7], re-quantization [7] or JPEG compression [8]. Moreover, to detect an image as adversarial, the probability vector from a classifier for an image \( \hat{x} \) and its transformed version, \( \phi(\hat{x}) \), can be compared with the \( L_1 \) norm as

\[
\| M(\hat{x}) - M(\phi(\hat{x})) \|_1 > \tau, \quad (4)
\]

where \( \tau \) is learned to accept a specific false-positive rate [7].

### III. Background

Adversarial images are crafted by constraining the added perturbation with, typically, an \( L_p \) norm between \( x \) and \( \hat{x} \). Attacks may constrain the total number of perturbed pixels \( (L_0) \) [3], the sum of magnitudes \( (L_1) \) [16], the Euclidean distance \( (L_2) \) [4], [15], or the maximum per-pixel variation \( (L_\infty) \) [1], [6], [12]. As the optimization of the perturbation, regularized by the norm, has no closed-form solution due to non-linear operations with the parameters, \( \theta \), and non-convex cost functions used to train the deep neural network classifiers, several adversarial attacks iteratively generate the adversarial image, \( \hat{x} \), as [12]:

\[
x_{n+1} = x_n + \delta_{x_n}, \quad (5)
\]

from \( x_0 = x \). The iteration process stops when a specific number of iterations, \( N \), is reached, i.e. \( x = x_{N+1} \), or when the misleading objective is achieved [3], [15]. \( N \) is typically chosen as a function of parameters linked to the preservation of image quality [3], [10]. Table II provides a comparative summary of adversarial attacks. Moreover, Figure I shows examples of the magnitude of perturbations generated by representative adversarial attacks, which are described next.

JSMA [3], a targeted attack, uses the \( L_0 \) norm and aims to identify and perturb by one intensity unit, at each iteration, the two most effective pixels. JSMA increases the probability of the target class, \( y_t \), compared with that of any other classes, using a saliency score, \( S(\cdot) \), which determines the effect of perturbing a pixel, \( x_i \in x \):

\[
S(x_i, y_t) = \begin{cases} 
0 & \text{if } \nabla_{x_i} M_t(x) < 0 \text{ or } \sum_{j \neq t} \nabla_{x_i} M_j(x) > 0 \\
\nabla_{x_i} M_t(x) \left\| \sum_{j \neq t} \nabla_{x_i} M_j(x) \right\|_1 & \text{otherwise,}
\end{cases}
\]

where \( \nabla_{x_i} M_t(x) \) is the gradient of the target class, \( y_t \), with respect to pixel \( x_i \). The attack iteratively generates the perturbation, until the adversarial image is classified as the target class or a specified number of pixels have been perturbed.

Carlini-Wagner (CW) [4], a targeted attack, maximizes the difference between the logarithmic probabilities of the target class and other classes using the \( L_2 \) norm. Three forms of selecting the target classes were used, namely uniform random selection, second- and least-likely cases.

DeepFool [15], an untargeted attack, crafts adversarial perturbations controlled by the \( L_2 \) norm. At each iteration, the adversarial perturbation is the orthogonal projection of the adversarial image from the previous iteration onto the closest linearized class boundary of \( M(\cdot) \). The final adversarial image, \( \hat{x} = x_{N+1} \), is the one that exceeds the closest class boundary as

\[
x_{N+1} = x_N + (1 + \eta) \delta x_N, \quad (7)
\]

where \( \eta \ll 1 \) and \( \delta x_N \) is the final adversarial perturbation. As DeepFool might generate adversarial images whose pixels exceed the allowed dynamic range, SparseFool [16], an untargeted attack similar to DeepFool, clips the pixel values within \([l, u]\), as

\[
\min \| x - x \|_1 \quad \text{s.t.} \quad y_s \neq y_t \quad \text{and} \quad 1 \leq x \leq u. \quad (8)
\]

This minimization is solved based on the low-mean curvature properties of the decision boundary of each image [7]. Note that, unlike DeepFool, SparseFool uses the \( L_1 \) norm, thus generating sparse perturbations.

The family of attacks based on FGSM generates the perturbation in the direction of the cost function, \( J_M(\cdot) \), in order to maximize the cost of remaining in the predicted class \( y_x \).
where $\nabla$ transformations, $\Phi$ the stronger the image degradation.

Let $L_{\text{Lab}}$ image in the larger misleading rate and the quality of the adversarial image: $E$ where $\epsilon$ to $J$.

The process can be reversed [1]. Random-FGSM (R-FGSM) [10], compromise the protection of the image, as the selection

We propose an iterative approach, Robust Private FGSM (RP-FGSM), that avoids overfitting by randomly selecting, at each iteration, a classifier to attack and a defense to evade. The proposed approach differs from E-FGSM [6] and DI-FGSM [13], which consider an ensemble of classifiers, and from EOT [5] and DI-FGSM [13], which use only non-defense transformations (e.g. rotations). Moreover, as in other FGSM-based attacks [12], RP-FGSM maintains image quality inherently by controlling the magnitude of the perturbation with the parameter $\epsilon$. The block diagram of RP-FGSM is shown in Figure 2.

Let $M = \{M_k(\cdot)\}_{k=1}^K$ be a set of $K \geq 1$ classifiers, and $\Phi = \{\phi_f(\cdot)\}_{f=0}^F$ be a set of $F$ defense transformations, where $\phi_0(\cdot)$ is the identity function (i.e. no transformation is applied to the input). For a classifier $M_k(\cdot)$ and transformation $\phi_f(\cdot)$, let $y_{x}^k$ and $y_{\phi_f(x)}^k$ be the predicted classes (Eq. 1) of the adversarial image, $x$, and the transformed adversarial image, $\phi_f(x)$, respectively. We generate the adversarial image, $x$, for $x$, whose true class is $y_x$, as

\begin{equation}
\hat{y}_x \neq y_x^k \text{ and } \hat{y}_x \neq y_{\phi_f(x)}^k \forall f, k.
\end{equation}

The process is initialized with $x_0 = x$ and ends with $x = x_{n+1}$. At each iteration, $n$, we randomly select a transformation $R(\Phi) \in \Phi$ and a classifier $R(M) \in M$, where $R(\cdot)$ is a function that randomly selects an element from a set.

We use the most effective defense transformations, namely re-quantization [7], median filtering [7] and JPEG compression [8]. Both re-quantization and JPEG compression have a quantization and rounding step, where the derivative, needed to compute the perturbation, is zero for almost every pixel, thus resulting in a null adversarial perturbation. We prevent

(untargeted) or to minimize the cost of predicting the targeted class $y_t$ (targeted):

\begin{equation}
\delta_x = \begin{cases}
\delta \text{ sign}(\nabla_x J_M(\theta, x, y_t)) & \text{untargeted attack}, \\
-\delta \text{ sign}(\nabla_x J_M(\theta, x, y_t)) & \text{targeted attack},
\end{cases}
\end{equation}

where $\delta$ controls the magnitude of the perturbation, $\text{sign}(\cdot)$ is the sign function and $\nabla_x J_M(\cdot)$ is the gradient of the cost function, $J_M(\cdot)$, showing the updated direction with respect to $x$. FGSM-based attacks use the $L_{\infty}$ norm and limit $x$ to the $\epsilon$-neighborhood of $x$ and within the dynamic range of an image at each iteration, as

\begin{equation}
\hat{x}_{n+1} = C_{x, \epsilon}(x_n + \delta x_n),
\end{equation}

where $C_{x, \epsilon}(\cdot)$ is a clipping function, defined as

\begin{equation}
C_{x, \epsilon}(x) = \min\{255, x + E, \max\{0, x - E, \hat{x}\}\},
\end{equation}

where $E = \{\epsilon\}^{W \times H \times C}$. The value of $\epsilon$ is a trade-off between the misleading rate and the quality of the adversarial image: the larger $\epsilon$, the higher the potential misleading rate but, also, the stronger the image degradation.

To evade defenses based on transformations and to preserve visual quality, Expectation Over Transformation (EOT) [5] optimizes the loss on the target class over a set of pre-defined 2D transformations, $\Phi_{2D}$, while minimizing the distance between the transformed original image and the transformed adversarial image in the $Lab$ color space [18].

\begin{equation}
\hat{x}_{n+1} = C_{x, \epsilon}(\hat{x}_n - \delta \text{ sign}(\nabla_x J_M(\theta, x_{2D}(\hat{x}_n), y_t) + \\
+ \lambda \|\mathcal{L}(\phi_{2D}(x_n)) - \mathcal{L}(\phi_{2D}(x_n))\|_2),
\end{equation}

where $\mathcal{L}(\cdot)$ is a function that converts images from the RGB to the $Lab$ color space, $\lambda$ controls the visual similarity, and $\phi_{2D}(\cdot) \in \Phi_{2D}$ is an image transformation chosen with uniform random probability at each iteration.

To increase the misleading rate with unseen classifiers, similarly to EOT, Diverse Input Fast Gradient Sign Method (DI-FGSM) [13] applies a random resizing followed by a padding transformation, with a pre-defined probability, on the adversarial image at each iteration.

To improve misleading performance with unseen classifiers, Ensemble FGSM (E-FGSM) [6] employs multiple classifiers, $M_k(\cdot)$, simultaneously when perturbing the creation:

\begin{equation}
\hat{x}_{n+1} = C_{x, \epsilon}(x_n - \delta \text{ sign}\left(\sum_{k=1}^K \nabla_x J_{M_k}(\theta, x_n, y_t)\right)),
\end{equation}

where $K \geq 1$. However, the use of all available classifiers at each iteration results in an overfitting that limits the ability of the adversarial image to mislead unseen classifiers [13].

Regarding the approach for selecting the target class, Least-likley FGSM (L-FGSM) [12] selects the least-likely predicted class. However, this systematic target class selection can compromise the protection of the image, as the selection process can be reversed [1]. Random-FGSM (R-FGSM) [10], E-FGSM [6] and EOT [5] randomly select the target class from all possible classes except the predicted class and the risk of reversibility is negligible. However, as the goal of the adversarial attacks is to hide the true class, which may not be the same as the predicted class (see Table II), this strategy can result in the selection of the true class as the target. Private-FGSM (P-FGSM) [11] avoids targeting the true class by discarding from the selection process the top predicted classes, which are more likely to contain the true class. The selection from the remaining classes is still random, thus limiting the risk of reversibility. Instead, untargeted approaches [12, 13] do not need the selection of a target class. However, knowledge of the true class is required for privacy protection.

IV. ROBUST AND PRIVATE FGSM

We aim to generate adversarial perturbations that protect the true class of images and thus protect the private information they contain from unwanted automatic inferences. These adversarial perturbations should be robust to defenses, be undetectable, preserve image quality, and mislead both seen and unseen classifiers.

We propose an iterative approach, Robust Private FGSM (RP-FGSM), that avoids overfitting by randomly selecting, at each iteration, a classifier to attack and a defense to evade. The proposed approach differs from E-FGSM [6] and DI-FGSM [13], which consider an ensemble of classifiers, and from EOT [5] and DI-FGSM [13], which use only non-defense transformations (e.g. rotations). Moreover, as in other FGSM-based attacks [12], RP-FGSM maintains image quality inherently by controlling the magnitude of the perturbation with the parameter $\epsilon$. The block diagram of RP-FGSM is shown in Figure 2.

Let $M = \{M_k(\cdot)\}_{k=1}^K$ be a set of $K \geq 1$ classifiers, and $\Phi = \{\phi_f(\cdot)\}_{f=0}^F$ be a set of $F$ defense transformations, where $\phi_0(\cdot)$ is the identity function (i.e. no transformation is applied to the input). For a classifier $M_k(\cdot)$ and transformation $\phi_f(\cdot)$, let $y_{x}^k$ and $y_{\phi_f(x)}^k$ be the predicted classes (Eq. 1) of the adversarial image, $x$, and the transformed adversarial image, $\phi_f(x)$, respectively. We generate the adversarial image, $x$, for $x$, whose true class is $y_x$, as

\begin{equation}
\hat{y}_x \neq y_x^k \text{ and } \hat{y}_x \neq y_{\phi_f(x)}^k \forall f, k.
\end{equation}

The process is initialized with $x_0 = x$ and ends with $x = x_{n+1}$. At each iteration, $n$, we randomly select a transformation $R(\Phi) \in \Phi$ and a classifier $R(M) \in M$, where $R(\cdot)$ is a function that randomly selects an element from a set.

We use the most effective defense transformations, namely re-quantization [7], median filtering [7] and JPEG compression [8]. Both re-quantization and JPEG compression have a quantization and rounding step, where the derivative, needed to compute the perturbation, is zero for almost every pixel, thus resulting in a null adversarial perturbation. We prevent

\begin{center}
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
ResNet18 & ResNet50 & AlexNet & DenseNet161 & \\
\hline
T1 & T5 & T1 & T5 & T1 & T5 & T1 & T5 \\
\hline
54.6 & 84.4 & 56.4 & 86.5 & 58.4 & 79.0 & 58.4 & 79.0 \\
\hline
\end{tabular}
\end{center}

TABLE II
CLASSIFICATION ACCURACY IN THE TEST SET OF THE PRIVATE PLACES365 DATASET FOR ResNet18, ResNet50, AlexNet AND DenseNet161. KEY - T1: TOP-1 CLASSIFICATION ACCURACY, T5: TOP-5 CLASSIFICATION ACCURACY.
the generation of a null perturbation when applying these transformations by approximating each pixel of the image $\hat{x}_n$ as $[\hat{x}_n^i] + ([\hat{x}_n^i] - [\hat{x}_n^i])^\delta$, where $[\cdot]$ represents the rounding operator to the nearest integer, and $i$ is the pixel index, whose derivative is unlikely to be zero \cite{eq19}. To avoid overfitting to certain transformations and parameters, the parameters of each transformation (i.e. the number of bits for re-quantization, the kernel size for median filter and the compression parameters for JPEG compression) are also chosen randomly from a pre-defined set of values \cite{eq13} (more details in Sec. V-B).

RP-FGSM can craft an adversarial image using targeted or untargeted attack by adding, at each iteration, the perturbation defined set of values \cite{eq13} (more details in Sec. V-B).

\begin{equation}
\delta_{\hat{x}_n} = \begin{cases} 
\delta \text{sign}(\nabla_x J_{M_t}(\theta, \phi_r(\hat{x}_n), y_\mathcal{K})) & \text{untargeted,} \\
-\delta \text{sign}(\nabla_x J_{M_t}(\theta, \phi_r(\hat{x}_n), y_t)) & \text{targeted,}
\end{cases}
\end{equation}

where $y_\mathcal{K}$ is the class predicted by the classifier $M_t(\cdot)$. Moreover, RP-FGSM maintains the perturbation within the $\epsilon$-neighborhood of $\hat{x}$ as

\begin{equation}
\hat{x}_{n+1} = \mathcal{C}_{\hat{x}, \epsilon}(\hat{x}_n + \delta_{\hat{x}_n}),
\end{equation}

where $\mathcal{C}_{\hat{x}, \epsilon}(\cdot)$ is the clipping function (Eq. \ref{eq11}).

Untargeted RP-FGSM crafts the adversarial perturbation in such a way that the predicted adversarial class moves farther away, in the decision space, from the most-likely predicted class for each classifier, which we assume to be in the neighborhood of the true class.

Targeted RP-FGSM selects the target class from a set that is more likely to exclude the true class, by leveraging the fact that the true class is often among the top classes predicted by a classifier \cite{eq11}. Moreover, we reduce the risk that an attack is reversed by randomly selecting the target class. Specifically, we exploit the prediction probability vectors $\{p_k^\mathcal{K}\}$ of the $K$ classifiers. Let $p^\mathcal{K}_x = (p^\mathcal{K}_i)_{i=1}^D$ contain the elements of $p^\mathcal{K}_x$ sorted in descending order. The target class is selected as

\begin{equation}
y_t = R \left( \bigcap_{k=1}^{K} \left\{ y_{j+1} : \sum_{i=1}^{j} p^k_i > \gamma, j \in \{1, ..., D - 1 \} \right\} \right),
\end{equation}

where $R(\cdot)$ randomly selects an element from the set of candidate target classes, obtained as the intersection of the subset of classes whose cumulative probability exceeds a threshold $\gamma \in [0, 1]$. The larger $\gamma$, the fewer the classes.

Figure 3 shows the influence of $\gamma$ on the number of classes from which the target class $y_t$ is selected.

After $N$ iterations, the attack uses, on average, each classifier $N/K$ times and each transformation $N/F$ times. The number of iterations $N$ might affect the quality of the image and therefore we choose it as a function of $\epsilon$ \cite{eq10} (see Sec. V-B).
V. Validation

A. Experimental setup

We compare the targeted and untargeted versions of the proposed attack, Robust Private FGSM (RP-FGSM), with eleven state-of-the-art methods, namely Jacobian-based Saliency Map Approach (JSMA) [3], Carlini-Wagner (CW) with $L_2$ norm [4], DeepFool [15], SparseFool [16], iterative untargeted FGSM (U-FGSM) [12], Random FGSM (R-FGSM) [10], Least-likely FGSM (L-FGSM) [12], Private FGSM (P-FGSM) [11], Expectation Over Transformation (EOT) [5], Diverse Input FGSM (DI-FGSM, targeted and untargeted versions) [13] and Ensemble FGSM (E-FGSM) [6]. JSMA, CW, DeepFool and SparseFool are run with the parameters proposed by their authors. The target class for JSMA is the one that introduces the smallest perturbation, whereas for CW it is the second most-likely predicted class. The target class is the least-likely predicted class for L-FGSM, chosen as described in Eq. 17 for P-FGSM and RP-FGSM, and, for the other targeted FGSM-based attacks, is selected randomly from the set of possible classes, excluding the predicted class of the original image. For untargeted FGSM-based attacks, we use the predicted class to craft the adversarial perturbation.

The Places365-Standard dataset [20], which has over 1.8 million images of 365 scene classes, is used for training the classifiers (see Table [1]). We use, as test set, the scene privacy subset of the validation set defined for the MediaEval 2018 Pixel Privacy Challenge [9], which comprises 60 private classes that are a subset of the classes of the Places365-Standard dataset [20]. The test set is composed of 3,000 images from each of the (60) private classes. We consider four state-of-the-art classifiers, namely ResNet [21] with 18 and 50 layers, AlexNet [22] and DenseNet161 [23], trained for scene classification [20]. Table [1] reports their classification accuracy on the test set. Adversarial attacks are performed on ResNet18, ResNet50 and AlexNet, whereas DenseNet161, the most accurate classifier, is only used as an unseen classifier for testing. This represents the most challenging setup, given that the largest and most accurate classifier was never seen by the adversarial attacks [6].

B. Implementation details

We compare all FGSM-based attacks with the same parameters proposed in Kurakin et al. [12]. The adversarial noise per iteration is $\delta = 1$, which corresponds to the smallest variation in an 8-bit image. We constrain the maximum perturbation magnitude by setting $\epsilon = 16$, to provide a trade-off between the misleading rate and image quality degradation [13]. For a fair comparison, we ensure that all FGSM-based attacks, either employing one classifier or an ensemble of classifiers (E-FGSM and DI-FGSM), perform the same number of forward/backward passes on the classifiers, with $N = \min(1.25 \epsilon, \epsilon + 4)$ [12] iterations. The number of iterations for RP-FGSM is $N = N \cdot K$, as only one classifier is used at each iteration (ensembles use all classifiers at each iteration). This ensures that all attacks that employ $K$ classifiers perform $K \cdot N$ forward/backward passes in total.

For EOT, we use as 2D transformations, $\Phi_{2D}$, scaling, translation, rotation, lightening, darkening and additive Gaussian noise. The specific parameters of the transformations are chosen randomly from pre-defined intervals as follows: scaling with a factor between 0.8 and 1.2; translation between -0.2 and 0.2 times $W$; rotation between -60 and 60 degrees; lightening/darkening between 0 and 13 intensity points; and Gaussian noise with zero mean and 25 variance. We set $\lambda = 0.5$ to achieve a misleading rate of approximately 90%, as suggested in [5]. For DI-FGSM, we employ random resizing followed by random padding with a probability of 50% at each iteration. Original images $x \in \mathbb{R}^{224 \times 224 \times C}$ are resized and padded to $r \times r \times C$ with $r$ randomly chosen within the range (200, 224).

The parameters for the defense transformations are: for re-quantization, 1 to 7 bits per color channel in steps of 1 [7]; for median filtering, squared kernel of dimensions 2, 3 and 5 [7]; and for lossy JPEG compression, quality parameters 25, 50, 75 and 100 [24, 25]. For RP-FGSM, we randomly and uniformly select the transformation to apply to the image at each iteration, by selecting the parameters (i.e. number of bits for re-quantization, kernel dimension for median filtering and compression factor for JPEG compression) from the same sets used for the defense. As Li et al. [11], in targeted attacks, we use $\gamma = 0.99$ for target class selection.

C. Performance measures

We consider as performance measures the misleading rate, detectability and image quality.

Misleading rate is the ratio between the number of adversarial images that mislead the classifier with respect to the target class and the total number of images, expressed as a percentage. The higher the misleading rate in top ranks, the better the privacy protection. We consider as ranks top-1 and top-5. The misleading rate is calculated for seen and unseen classifiers, and with and without defenses.

Detectability is the percentage of correctly detected adversarial images with respect to the total number of images. We distinguish between adversarial images and original images by learning a threshold, $\tau$, for each defense (re-quantization, median and JPEG compression) by accepting 5% of the original (training) images as adversarial (i.e. 5% false positive detection rate) (Eq. 4) [7].

Image quality is evaluated with Blind Referenceless Image Spatial Quality Evaluator (BRISQUE) [26], a no-reference...
measure; and Peak-Signal-to-Noise Ratio (PSNR) and Most Apparent Distortion (MAD) [27], two full-reference measures. BRISQUE [26] quantifies distortions and unnaturalness. The lower the BRISQUE score, the better the image quality. PSNR quantifies the pixel-by-pixel difference between two images. The higher the PSNR, the smaller the added perturbation (and therefore, the better the image quality). MAD aims at measuring the perceived quality under different levels of distortion [27]. The lower the MAD score, the higher the image quality.

D. Evaluation of privacy protection

Figure 4 compares the results of RP-FGSM with those of the eleven considered state-of-the-art attacks. For visualization purposes, all metrics are normalized using their known ranges to 0% and 100%, where a higher value is more desirable. We report undetectability, the inverse of detectability; the top-1 misleading rate with seen (unseen) classifiers obtained with ResNet50 (DenseNet161); and image quality in terms of naturalness, BRISQUE. We report the results of the attacks when using the classifier (or combination of classifiers) with the highest misleading rate: ResNet50 for attacks that use a single classifier, and the combination of ResNet18, ResNet50 and AlexNet for attacks that use more than one classifier.

The top row of Figure 4 shows the trade-off between misleading seen (unseen) classifiers with and without defenses. The proposed attack outperforms the other attacks in these two plots (i.e. is nearer to the top-right corner of the plot). The second row of Figure 4 shows the relationship between image quality and undetectability. We can observe that all attacks obtain similar image quality, whereas attacks can be divided into two groups for the undetectability, those below 40% (CW, DeepFool, R-FGSM, L-FGSM, P-FGSM, EOT, E-FGSM, and DI-FGSM targeted), and those above 40% (JSMA, SparseFool, U-FGSM, and DI-FGSM untargeted). The proposed attack, RP-FGSM, in both targeted and untargeted versions, is within the group of less detectable attacks.

Table V-D summarizes all performance measures.

For seen classifiers without defenses, JSMA, CW, DeepFool and SparseFool have the lowest misleading rate (lower than 32.1% in top-5) as they craft their perturbation towards the closest class to the predicted class and the resulting class of the adversarial image could be the true class that we aim to protect (see Table II). Instead, targeted (untargeted) FGSM-based attacks have higher misleading rates because they move the prediction of the adversarial image closer to (farther from) the target (predicted) class at each iteration. The misleading rates of EOT are only between 67.4% and 72.7% in top-5, showing a limited ability to mislead the classifier when no transformation is applied. The misleading rates of untargeted FGSM-based attacks, such as U-FGSM and DI-FGSM, are above 83.4% in top-5, whereas those of RP-FGSM are above 86.9% when attacking a single classifier. Targeted FGSM-based attacks such as L-FGSM and P-FGSM obtain the highest misleading rates (above 95% in top-5). Similar results are obtained by attacks that use an ensemble of classifiers (E-FGSM and DI-FGSM). The target (untargeted) version of RP-FGSM obtains comparable results to the other FGSM-based attacks, i.e. above 97.8% (86.9%) in the top-5.

For an unseen classifier, JSMA, CW, DeepFool, SparseFool and EOT obtain low misleading rates (under 21.6% in the top-5). Similarly to the above results for seen classifiers, targeted FGSM attacks such as R-FGSM, L-FGSM, P-FGSM and E-FGSM (attacking a single classifier) obtain a higher performance, with misleading rates between 15.5% and 88.7% in the top-5. The highest misleading rates are provided by DI-FGSM and RP-FGSM when using three classifiers for generating the attack. In their targeted versions, they obtain misleading rates of 65.0% and 64.6% in the top-5, respectively, while the untargeted versions reach 76.0% and 82.6%, respectively. Adversarial attacks that combine classifiers obtain the highest misleading rates in unseen classifiers. Targeted DI-FGSM and RP-FGSM have comparable results (0.4 percentage points difference), while untargeted RP-FGSM outperforms untargeted DI-FGSM by more than 6 percentage points, thus indicating that the random selection of classifiers and transformations is an effective strategy for privacy protection.

For seen classifiers with defenses, JSMA, CW, DeepFool and SparseFool have low misleading rates (less than 21.2% in the top-5). Targeted attacks such as R-FGSM, L-FGSM, P-FGSM, EOT and E-FGSM (attacking one classifier) obtain much lower misleading rates compared to when no defenses are employed (19.8% - 53.4% in the top-5). Also in this case, untargeted FGSM-based attacks perform the best in the top-5 (above 47.4% for U-FGSM and above 59.3% for DI-FGSM). RP-FGSM outperforms all attacks when three classifiers are employed for crafting the adversarial images, with misleading rates in the top-5 between 59.6% and 88.0% when targeted, and between 70.3% and 91.7% when untargeted. When attacking three classifiers, the misleading rate of E-FGSM drops by 61.6 percentage points (35.2% from 96.8%) when defenses
| Att. Classifier | Misleading † | Misleading with defenses † | Det. | Image quality |
|----------------|-------------|---------------------------|------|---------------|
|                | R18         | R50                       | A    | R18           |
|                | 85.3±15.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 43.0±12.0   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 84.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 44.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 45.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 46.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 47.5±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 48.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 49.7±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 50.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 52.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 53.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 54.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 55.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 56.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 58.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 59.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 60.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 61.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 62.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 64.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 65.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 66.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 67.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 68.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 70.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 71.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 72.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 73.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 74.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 76.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 77.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 78.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 79.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 80.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 82.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 83.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 84.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 85.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 86.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 88.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 89.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 90.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 91.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 92.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 94.0±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 95.2±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 96.4±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 97.6±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 98.8±13.8   | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |
|                | 100.0±13.8  | 44.6±13.8                 | 52.6±7.3 | 42.4±13.8     |

**TABLE III**

Privacy protection results in the test split of Private Places365 dataset for misleading rate, with and without defenses, detectability and image quality. Measurements are reported as average of the dataset. Standard deviation for image quality is shown in brackets. Gray shading indicates unseen classifier. Bold font indicates the best performing attack for a given performance measure. KEY – Att.; attack; R18: ResNet18; R50: ResNet50; A: AlexNet; DN: DenseNet161; Det.: Detectability; BRISQUE: Blind Referenceless Image Spatial Quality Evaluation; PSNR: Peak-Signal-to-Noise Ratio; MAD: Most Apparent Distortion; T1: Top-1 Misleading rate; T5: Top-5 Misleading rate; JSMA: Jacobian-based Saliency Map Attack; CW: Carlini-Wagner; DF: DeepFool; SF: SparseFool; U-FG: Untargeted FGSM; R-FG: Random FGSM; L-FG: Least-Likely FGSM; P-FG: Private FGSM; EOT: Expectation Over Transformation; DI-FG: Diverse Input FGSM; E-FG: Ensemble FGSM; RP-FGSM: Robust Private FGSM, the proposed attack; T: Targeted attack; U: Untargeted attack.
are applied. Instead, RP-FGSM untargeted only drops by 3.3 percentage points (87.7% from 91.0%).

When evaluating misleading unseen classifiers with defenses, JSMA, CW, DeepFool and SparseFool still obtain the lowest misleading rates (lower than 21.2% in top-5). Similarly, FGSM-based attacks such as R-FGSM, L-FGSM, P-FGSM, EOT and E-FGSM obtain very low misleading rates (under 27.1% in the top-5), while DI-FGSM, when attacking three classifiers, obtains higher misleading rates with values of 40.1% (targeted) and 53.7% (untargeted). Also in this case, the proposed RP-FGSM outperforms all other attacks, obtaining a misleading rate in the top-5 of 55.1% (targeted) and of 77.0% (untargeted). This confirms that untargeted attacks exhibit a higher degree of privacy protection with unseen classifiers, both with and without defenses.

Table IV compares predicted probabilities of sample adversarial images. We observe that most of the adversarial attacks are unsuitable for privacy protection as the true class might be within the top-5. DI-FGSM and RP-FGSM are the only attacks capable of removing the true class from the top-5 for all classifiers in both seen (ResNet18, ResNet50 and AlexNet) and unseen (DenseNet161) settings. This is mainly due to the fact that the attacks do not overfit the perturbations to a specific classifier and that the crafted perturbations generate adversarial predicted classes that are far from the predicted class in the decision space.

Targeted attacks are in general more detectable than their untargeted versions, as reaching the target class often means crafting a larger perturbation. R-FGSM, L-FGSM, P-FGSM, EOT and E-FGSM are highly detectable, with values above 93.2%, while U-FGSM obtains detection rates below 58.1%. The detectability of targeted DI-FGSM is between 82.8% and 92.2%, and that of targeted RP-FGSM between 55.9% and 79.0% (RP-FGSM is less detectable than DI-FGSM in this setting). Untargeted DI-FGSM obtains detectability between 41.0% and 53.2%, while untargeted RP-FGSM between 41.7% and 57.6%, thus DI-FGSM is slightly less detectable than RP-FGSM. JSMA and SparseFool have the lowest detection rates, with values under 42.1%.

Attacks with a sparse perturbation, such as JSMA and SparseFool, have the highest image quality results (PSNR values above 38.4 dB). However, sparse perturbations are often of high magnitude (Figure I) and, therefore, highly noticeable (Table IV). This effect is reflected by the poor BRISQUE score, above 51.0 points, for JSMA and SparseFool. Attacks with dense perturbations obtain a lower PSNR than sparse ones, whose accumulated variation of intensity across the whole image is smaller. However, dense perturbations might be more desirable for privacy protection, as they are less visible and their per-pixel intensity variation is lower than that of sparse attacks. FGSM-based attacks obtain consistently better BRISQUE scores than non-FGSM based attacks. FGSM-based attacks obtain around 44 BRISQUE score and non-FGSM based attacks obtain around 5 points more (the lower the better). EOT, which has an image quality regularization in its modeling (see Eq. [12]), is the FGSM-based attack with the highest PSNR results (above 42.3 dB) and the lowest MAD score (below 3.6 points). However, its BRISQUE scores are higher (i.e. worse) than those of other FGSM-based attacks, thus indicating a lower image naturalness. When the number of classifiers used to create the adversarial images increases, a higher level of perturbations is required, which results in a decrease of image quality. For instance, we see a decrease of about 3-4 dBs when DI-FGSM, E-FGSM and RP-FGSM
employ three classifiers instead of only one. However, FGSM-based attacks can obtain the desired trade-off between misleading rate and image quality by tuning the parameter $\epsilon$ (Eq. (11)), a desirable feature for applications such as privacy protection (see the sensitivity analysis in Sec. [V-E]). RP-FGS obtains quality scores comparable to other FGSM-based attacks in terms of BRISQUE and PSNR. For instance, when attacking three classifiers, E-FGS obtains BRISQUE score 41.8 and PSNR 33.8 dB, whereas RP-FGS obtains BRISQUE score 33.4 (the lower the better) and PSNR 29.6 dB (the higher the better). However, among the FGSM methods for three classifiers, E-FGS obtains the best MAD score 19.8, whereas RP-FGS obtains the worst MAD score 40.5 (the lower the better).

E. Ablation and sensitivity analysis

We present two analyses of RP-FGS, namely an ablation study to validate the effectiveness of the random selection of classifier and validation, and of different number of iterations; and a sensitivity analysis on the parameter $\epsilon$.

The ablation study compares the effect of using a randomly selected transformation with respect to not using any transformations, and the effect of using randomly selected classifiers with respect to using an ensemble of classifiers at each iteration. Furthermore, we evaluate the effect of the number of iterations, $N$, when attacking multiple classifiers. For a fair comparison, all these experiments follow the same sequence of transformations, classifiers and target class, whenever applicable, by fixing the random seed. We report the variation of performance measures, namely misleading rates with and without defense, and detectability, for both targeted and untargeted RP-FGS.

Table [V] shows the effect of using random transformations with respect to not using any transformation. When employing a single classifier, using transformations generally improves the misleading and decreases the detectability. In particular, the detectability decreases by an average of 19.4 (5.5) percentage points in the targeted (untargeted) version. When using an ensemble of classifiers, the misleading generally increases for both targeted and untargeted. For the untargeted version, the detectability increases by an average of 9.5 percentage points, whereas for the targeted one, increases by an average of 15.7 percentage points. We argue that using an ensemble of classifiers generates overfitted perturbations that successfully mislead the classifier, but are more detectable. When using a random classifier at each iteration, as we propose, the misleading with defense increases in both targeted and untargeted versions. Moreover, the detectability in the untargeted attack decreases, contrary to when using an ensemble of classifiers. This experiment shows that introducing random transformations improves the performance, especially when a random classifier is employed at each iteration.

Table [VI] compares the effect of using a randomly selected classifier with respect to using an ensemble of classifiers, at each iteration. We report the results when not using any transformation, for reference, and also with a random selection of transformations. When random transformations are employed in the generation, the use of a randomly selected classifier at each iteration increases, in general, the misleading rate in the targeted version by a maximum of 4.2 percentage points in top-1 with an unseen classifier ( DenseNet ). Also, detectability decreases by 12.5 (5.2) percentage points in the targeted (untargeted) version. Some instances of misleading rate decreases are observed, especially in the untargeted version.

Table [VII] reports how results vary using a random selection of the classifier with respect to an ensemble of classifiers, with varying numbers of iterations. We report the performance measures with the proposed number of iterations (i.e. 40, with two classifiers; and 60, with three classifiers), which allows the proposed attack to perform the same number of forward and backward passes on the classifiers compared to an ensemble FGSM (e.g. E-FGS). We also study the effect of the number of iterations on the performance measures by considering fewer iterations (i.e. 20). For an equal number of uses of a classifier, the random selection of classifier improves the misleading of unseen classifiers by 16.4 percentage points in the top-5 (when using three classifiers), and decreases the detectability by 28.7 percentage points. Moreover, when
TABLE VI
IMPACT ON MISLEADING AND DETECTABILITY OF USING A RANDOM SELECTION OF CLASSIFIERS WITH VARYING NUMBER OF ITERATIONS IN CRAFTING ADVERSARIAL PERTURBATIONS (VALUES ARE THE AVERAGE OF THE SCORE OBTAINED WHEN USING AN ENSEMBLE OF CLASSIFIERS [6]).

| Trans. | Classifier | Targeted | Misleading ↑ | Misleading with defense ↑ | UnTargeted | Misleading ↑ | Misleading with defense ↑ | Det. ↓ |
|--------|------------|----------|--------------|--------------------------|------------|--------------|--------------------------|-------|
|        |            | R18+R50  | T1 T5        | T1 T5                    | R18+R50    | T1 T5        | T1 T5                    |       |
| R18+R50 | N/A       | 0.0      | 0.0          | 0.0                      | 0.0        | 0.0          | 0.0                      |       |
| R18+R50 | R18       | 0.0      | 0.0          | 0.0                      | 0.0        | 0.0          | 0.0                      |       |
| R18+R50 | R50       | 0.0      | 0.0          | 0.0                      | 0.0        | 0.0          | 0.0                      |       |
| R18+R50 | R18+R50   | 0.0      | 0.0          | 0.0                      | 0.0        | 0.0          | 0.0                      |       |
|        |            | Average  | 0.0          | 0.0                      | 0.0        | 0.0          | 0.0                      |       |

TABLE VII
IMPACT ON MISLEADING AND DETECTABILITY OF USING A RANDOM SELECTION OF CLASSIFIERS WITH VARYING ITERATIONS IN CRAFTING ADVERSARIAL PERTURBATIONS (VALUES ARE THE AVERAGE OF THE SCORE OBTAINED WHEN USING AN ENSEMBLE OF CLASSIFIERS [6]).

| Trans. | Classifier | Misleading ↑ | Misleading with defense ↑ | Det. ↓ |
|--------|------------|--------------|---------------------------|-------|
|        |            | R18+R50      | T1 T5                     |       |
|        |            | Average      | 0.0                       |       |

F. Runtime analysis

We compare the runtime of each attack in an Ubuntu 18.04.3 server equipped with an NVIDIA Tesla V100 GPU, using a random subset of 300 images. The implementations are in Python using the PyTorch library [28]. Figure 6(a) shows that JSMA, CW and SparseFool are the slowest attacks taking on average 52.14, 9.87 and 4.42 seconds per image, respectively, when attacking ResNet50. DeepFool and FGSM-based attacks have a similar runtime performance with an average under 0.7 seconds per image. Figure 6(b) shows how the runtime of RP-FGSM increases when the number of attacked classifiers increases (similarly to DI-FGSM and E-FGSM). The major factor affecting the runtime is the forward/backward passes on the classifier.

VI. Conclusion

We presented RP-FGSM, a Robust and Private Fast Gradient Sign Method that is designed to mislead seen and unseen classifiers with and without known defenses. RP-FGSM has better performance than other state-of-the-art adversarial attacks for privacy protection in the Private Places365 dataset, especially for unseen classifiers and when defenses are applied. The key for this performance is the random selection of a defense transformation and a classifier at each iteration, which prevents the crafted perturbation from overfitting to a
particular classifier or defense. As future work, we will extend the validation to tasks beyond scene classification.

ACKNOWLEDGMENT

We thank the Alan Turing Institute (EP/N510129/1), which is funded by the EPSRC, for its support through the project PRIMULA.

REFERENCES

[1] C. Y. Li, A. S. Shamsabadi, R. Sanchez-Matilla, R. Mazzon, and A. Cavallaro, “Scene privacy protection,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, May 2019, pp. 2502–2506.

[2] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in Proceedings of the International Conference on Learning Representations (ICLR), San Diego, USA, May 2015.

[3] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), Saarbrücken, Germany, March 2016, pp. 372–387.

[4] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proceedings of the Symposium on Security and Privacy (S&P), San Jose, California, USA, May 2017, pp. 39–57.

[5] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, “Synthesizing robust adversarial examples,” in Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, July 2018, pp. 284–293.

[6] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” in Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada, April 2018.

[7] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” in Proceedings of the Network and Distributed Systems Security symposium (NDSS), San Diego, California, USA, February 2018.

[8] Z. Liu, Q. Liu, T. Liu, Y. Wang, and W. Wen, “Feature distillation: DNN-oriented JPEG compression against adversarial examples,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, June 2019, pp. 860–868.

[9] “Pixel privacy task, MediaEval 2018.” http://www.multimediaeval.org/mediaeval2018, 2018, [Last accessed March 2020].

[10] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” in Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, April 2017.

[11] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada, April 2018.

[12] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Proceedings of the International Conference on Learning Representations (ICLR) Workshop Track, Toulon, France, April 2017.

[13] C. Xie, Z. Zhang, J. Wang, Y. Zhou, Z. Ren, and A. Yuille, “Improving transferability of adversarial examples with input diversity,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, June 2019, pp. 2730–2739.

[14] Y. Dong, T. Pang, H. Su, and J. Zhu, “Evaing defenses to transferable adversarial examples by translation-invariant attacks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, June 2019, pp. 4312–4321.

[15] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in Proceedings
of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, USA, June 2016, pp. 2574–2582.

[16] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, “SparseFool: a few pixels make a big difference,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, June 2019, pp. 9087–9096.

[17] A. Fawzi, S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto, “Empirical study of the topology and geometry of deep networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, Utah, USA, June 2018, pp. 3762–3770.

[18] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao, “Statistics of cone responses to natural images: implications for visual coding,” Journal of the Optical Society of America (JOSA), vol. 15, no. 8, pp. 2036–2045, August 1998.

[19] R. Shin and D. Song, “JPEG-resistant adversarial images,” in Proceedings of the Advances in Neural Information Processing Systems (NIPS) workshop, Long Beach, California, USA, January 2017.

[20] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 40, no. 6, pp. 1452–1464, June 2018.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, USA, June 2016, pp. 770–778.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, USA, December 2012, pp. 1097–1105.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE Conference on computer vision and pattern recognition (CVPR), Honolulu, Hawaii, July 2017, pp. 4700–4708.

[24] G. Dziugaite, Z. Ghahramani, and D. Roy, “A study of the effect of JPEG compression on adversarial images,” arXiv preprint arXiv:1608.00853, August 2016.

[25] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, “Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression,” arXiv preprint arXiv:1705.02900, May 2017.

[26] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference image quality assessment in the spatial domain,” IEEE Transactions on Image Processing (TIP), vol. 21, no. 12, pp. 4695–4708, Dec. 2012.

[27] E. C. Larson and D. M. Chandler, “Most apparent distortion: full-reference image quality assessment and the role of strategy,” Journal of Electronic Imaging, vol. 19, no. 1, pp. 1 – 21, January 2010.

[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in PyTorch,” in Proceedings of the Advances in Neural Information Processing Systems Workshop (NIPS-W), Long Beach, California, USA, December 2017.