BAR-ILAN UNIVERSITY

Using Explainability To Detect Adversarial Examples

Ohad Amosy

Submitted in partial fulfillment of the completion course for Ph.D. of the

Department of Computer Science, Bar-Ilan University

Ramat Gan, Israel 2019

This work was carried out under the supervision of

Prof. Gal Chechik and Prof. Ely Porat

The Brain Research Center and Department of Computer Science, Bar-Ilan University.

Acknowledgement

I would like to thank my advisors, Prof. Gal Chechik and Prof. Ely Porat, for their guidance, encouragement and advice all over the way. Their support played a vital role in the creation of this work. Thank you.

I would like to thank my parents and my family, for their support and love through all my life. I couldn’t have done it without you.

Contents

List of Figures

Abstract

Deep learning models are often sensitive to adversarial attacks, where carefully-designed input samples can cause the system to produce incorrect decisions. Here we focus on the problem of detecting attacks, rather than robust classification, since detecting that an attack occurs may be even more important than avoiding misclassification. We build on advances in explainability, where activity-map-like explanations are used to justify and validate decisions, by highlighting features that are involved with a classification decision. The key observation is that it is hard to create explanations for incorrect decisions.

We propose EXAID, a novel attack-detection approach, which uses model explainability to identify images whose explanations are inconsistent with the predicted class. Specifically, we use SHAP, which uses Shapley values in the space of the input image, to identify which input features contribute to a class decision. Interestingly, this approach does not require to modify the attacked model, and it can be applied without modeling a specific attack. It can, therefore, be applied successfully to detect unfamiliar attacks, that were unknown at the time the detection model was designed.

We evaluate EXAID on two benchmark datasets CIFAR-10 and SVHN, and against three leading attack techniques, FGSM, PGD and C&W. We find that EXAID improves over the SoTA detection methods by a large margin across a wide range of noise levels, improving detection from $\sim 70\%$ to over 90% for small perturbations.

Chapter 1

Introduction

Machine learning systems can be tricked to make incorrect decisions when presented with samples that were slightly perturbed, but in special, adversarial ways [Szegedy *et al.*, 2013]. This sensitivity, which was widely studied, can hurt networks regardless of the application domain and can be applied without knowledge of the model [Papernot *et al.*, 2017]. Detecting such adversarial attacks is currently a key problem in machine learning.

To motivate our approach, consider how most conferences decide on which papers get accepted for publication. Human classifiers, known as reviewers, make classification decisions, but unfortunately, these are notoriously noisy. To verify that their decision is sensible, reviewers are also asked to explain and justify their decision. Then, a second classifier, known as an area-chair or an editor, examines the classification, together with the explanation and the paper itself, to verify that the explanation supports the decision. If the justification is not valid, the review may be discounted or ignored.

In this work, we build on a similar intuition: Explaining a decision can reduce misclassification. Clearly, the analogy is not perfect, since unlike human reviewers, for deep models we do not have trustworthy methods to provide high-level semantic explanations of decisions. Instead, we study below the effect of using the wider concept of explanation on detecting incorrect decisions, and in particular given adversarial samples that are designed to confuse a classifier. The key idea is that different classes have different explaining features and that by probing explanations, one can detect classification decisions that are inconsistent with the explanation. For example, if an image is classified as a dog, but has an explanation that gives high weight to a striped pattern, it is more likely that the classification is incorrect.

We focus here on the problem of *detecting* adversarial samples, rather than developing a system that provides *robust* classifications under adversarial attacks. This is because in many cases we are interested to detect that an attack occurs, even if we cannot automatically correct the decision.

The key idea in detecting adversarial attacks is to identify cases where the network behaves differently than when presented with untainted inputs, and previous methods focused on various different aspects of the network to recognize such different behaviors Lee *et al.* [2018]; Ma *et al.* [2018]; Liang *et al.* [2018]; Roth *et al.* [2019]; Dong *et al.* [2019]; Katzir and Elovici [2018]; Xu *et al.* [2017]. To detect these differences, here we build on recent work in explainability Lundberg and Lee [2017a]. The key intuition is that explainability algorithms are designed to point to input features that are the reason for making a decision. Even though leading explainability methods are still mostly based on high-order correlations and not necessarily identify purely causal features, they often yield features that people identify as causal [Lundberg and Lee, 2017b]. Explainability therefore operates directly against the aim of adversarial methods, which perturb images in directions that are not causal for a class. The result is that detection methods based on explainability holds the promise to work particularly well with adversarial perturbations that lead to nonsensical classification decisions.

There is a second major reason why using explainable features for adversarial detection is promising. Explainable features are designed to explain the classification decision of a classifier trained on non-modified (normal) data. As a result, they are independent of any specific adversarial attack. Some previous methods are based on learning the statistical abnormalities of the added perturbation. This makes them sensitive to the specific perturbation characteristics, which change from one attack method to another, or with change of hyperparameters. Instead, explainability models can be agnostic of the particular perturbation method.

The challenge in detecting adversarial attacks becomes more severe when the perturbations of the input samples are small. Techniques like C&W Carlini and Wagner [2017b] can adaptively select the noise level for a given input, to reach the smallest perturbation that causes incorrect classification. It is therefore particularly important to design detection methods that can operate in the regime of small perturbations. Explanation-based detection is inherently less sensitive to the magnitude of the perturbation, because it focuses on those input features that explain a decision for a given class.

In this work we describe an EXAID (EXplAIn-then-Detect), an explanation-based method to detect adversarial attacks. It is designed to capture low-noise perturbations from unknown attacks, by building an explanation model per-class that can be trained without access to any adversarial samples.

Our novel contributions are as follows:

- We describe a new approach to detect adversarial attacks using explainability techniques.
- We study the effect of negative sampling techniques to train such detectors.
- We also study the robustness of this approach in the regime of low-noise (small perturbations).
- We show that the new detection provides state-of-the-art defense against the three leading attacks (FGSM, PGD, CW) both for known attacks and in the setting of detecting unfamiliar attacks.

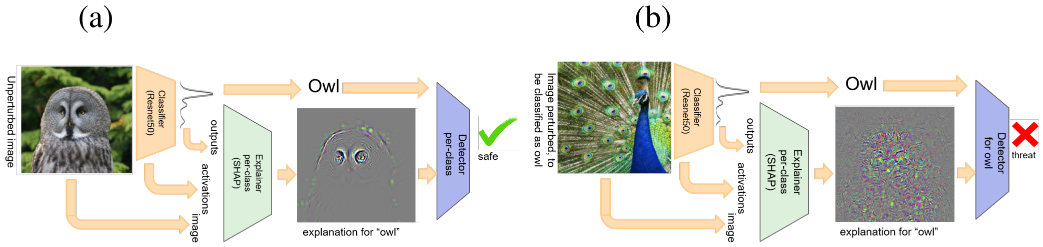


Figure 1.1: **Illustration of EXAID**. First, an image is classified by a standard image classification system like ResNet. Then, an explanation is created based on the image, the network activations and the network output. Finally, a detector checks if the generated explanation is consistent with the predicted label. (a) An image of an owl is correctly classified, and the produced explanation is consistent with the label “owl”. (b) An image of peacock is perturbed and used as an attack. It is falsely classified as an owl, and is detected as adversarial because its explanation is inconsistent with the predicted label.

Chapter 2

Related work

2.1 Explainable AI

Due to the high non-linearity and nested architectures of deep neural networks, it is challenging to intuitively understand how does a deep neural network arrives at a specific decision for a given input. This significantly impairs the use of deep neural networks for sensitive tasks, where black-box predictions cannot be trusted by default, and the ability to explain the result is required.

In the past few years, improving deep neural network interpretability has been an intensive research area. New methods provided explanations at either model-level [Karpathy *et al.*, 2015; Sabour *et al.*, 2017; Zhang *et al.*, 2018a], or instance-level [Dabkowski and Gal, 2017; Fong and Vedaldi, 2017; Ribeiro *et al.*, 2016]. In this work, we will focus on instance-level interpretability because our goal is to detect adversarial attacks on specific instances.

Interpretability methods can be obtained in two ways. The first way is designing interpretable models [Sabour *et al.*, 2017; Zhang *et al.*, 2018a; Chen *et al.*, 2019; Wang *et al.*, 2017]. These models are generally less accurate than non-interpretable models, which leads to a trade-off between model complexity and therefore accuracy and model interpretability. The second way is extracting post-hoc interpretations [Simonyan *et al.*, 2013; Smilkov *et al.*, 2017; Sundararajan *et al.*, 2017; Zhou *et al.*, 2016; Selvaraju *et al.*, 2017; Ribeiro *et al.*, 2016; Lundberg and Lee, 2017b; Shrikumar *et al.*, 2017] which does not require modifying model architectures or parameters, thereby leading to higher prediction accuracy. In this work, we will consider post-hoc interpretations because our goal is to create a robust defense that will be agnostic to the defended model.

We can divide post-hoc interpretation methods into two types: (1) Class activation map [Zhou *et al.*, 2016; Selvaraju *et al.*, 2017] - which produces a class-discriminative localization map. (2) Pixel sensitivity map [Simonyan *et al.*, 2013; Smilkov *et al.*, 2017; Sundararajan *et al.*, 2017; Shrikumar *et al.*, 2017; Lundberg and Lee, 2017b]- which uses calculations with gradients to assign importance scores to individual pixels toward explaining the classification of an input.

2.2 Adversarial attacks

The literature on adversarial attacks is vast and includes different attack scenarios and different ways to construct the adversarial examples. Basic attacks like Goodfellow *et al.* [2014] let the attacker manipulate the image without any constraints except the total amount of noise the attacker will add. A more challenging attack scenario is a *real world attack* scenario Papernot *et al.* [2017]. In this scenario, the attacker's ability to influence the model's input is only through the physical scene while taking the image, without any ability to change the digital image. This is a much more difficult setup due to the inability to control all the physical environment, like lighting conditions, photography angles, etc. Therefore, unlike the digital case, the attacker cannot determine the value of each pixel individually at will [Brown *et al.*, 2017; Eykholt *et al.*, 2018; Pautov *et al.*, 2019; Kurakin *et al.*, 2016a]. Another challenging attack scenario is when the attacker can change only a specific patch of the image [Brown *et al.*, 2017; Subramanya *et al.*, 2019] or even only one pixel [Su *et al.*, 2019]. In this work, we gave the attacker the ability to manipulate directly the digital image.

We focus here on three high-performing adversarial attacks which we use in our experiments. Each of the three represents a group of attacks that share the same main idea.

Fast Gradient Sign Method (FGSM). This attack by [Goodfellow *et al.*, 2014] creates a perturbation by ”moving” an example one step in the direction of the gradient. Let c be the true class of x and $J(C, x, c)$ be the loss function used to train our deep neural network C. The perturbation is computed as a sign of the model's loss function gradient $\Delta x = \epsilon * sign(\nabla_x J(C, x, c))$, where ϵ ranging from 0.0 to 1.0. The parameter ϵ controls the perturbation magnitude and can be thought of as the noise-level of the adversarial sample.

Projected Gradient Decent (PGD). Madry *et al.* [2017] suggested to improve FGSM, in the following way. One can interpret FGSM as a one-step scheme for maximizing the inner part of the saddle point formulation. A more powerful adversary will be a multi-step variant, which essentially applies projected gradient descent on the negative loss function $x^{t+1} = x^t + \epsilon * sign(\nabla_x J(C, x, c))$ while $x^0 = x$.

Carlini and Wagner (C&W). Carlini and Wagner [2017b] employed an optimization algorithm to seek the smallest perturbation that enables an adversarial example to fool the classifier. Given a neural network F with logits Z, the attack uses gradient descent to craft the adversarial example x' by solving $min||x' - x||_2^2 + c * l(x')$ where the loss function l is defined as $l(x') = max(max_i\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa)$. The difference max $max_i\{Z(x')_i : i \neq t\} - Z(x')_t$ is used to compare the target class t with the next-most-likely class. However, this is minimized when the target class is significantly more likely than the second most likely class, which is not a property we want. This is fixed by taking the maximum of this quantity with $-\kappa$ which controls the confidence of the adversarial examples. When $\kappa = 0$, the adversarial examples are called *low-confidence adversarial examples* and are only just classified as the target class. As κ increases, the model classifies the adversarial examples as increasingly more likely, this is called *high-confidence adversarial examples*. As showed in [Carlini and Wagner, 2017a], this attack is considered to be one of the most powerful attacks and therefore is a common baseline.

When designing attacks, previous studies took into account various factors: the probability that the attack is successful, the effect on the appearance of a perturbed image, and the time it takes to run the attack. The above three methods prioritize these aspects differently, reaching different trade-off operating points. Specifically, FGSM is usually faster and the C&W attack yields less-visible perturbation of the input images.

2.3 Detecting adversarial attacks

At a glance, defense methods are divided into three main methods:

1. Pre-process the input in order to break the effect of the adversarial perturbation [Prakash *et al.*, 2018; Gu and Rigazio, 2014; Osadchy *et al.*, 2017; Das *et al.*, 2017].

2. Train robust models that will be able overcome the existence of an adversarial perturbation and make a correct classification. [Goodfellow *et al.*, 2014; Kurakin *et al.*, 2016b; Tramèr *et al.*, 2017]

3. Detect adversarial examples [Liang *et al.*, 2018; Xu *et al.*, 2017; Ma *et al.*, 2018; Lee *et al.*, 2018; Katzir and Elovici, 2018; Pang *et al.*, 2018; Roth *et al.*, 2019].

Here we focus on the problem of detecting attacks, rather than robust classification, since detecting that an attack occurred may be even more important than avoiding misclassification. Several previous techniques have been proposed to detect adversarial examples. Liang *et al.* [2018] measured the effect of quantization and smoothing of the image on the network classification, both parameterized as a function of image entropy. Similarly, Xu *et al.* [2017] suggested to reduce the degrees of freedom of the input space by applying transformations like quantization and smoothing, and then comparing the model predication before and after each transformation. Ma *et al.* [2018] measured the characteristics of the region surrounding a reference example. Lee *et al.* [2018] models the distribution of activations at the hidden layers of the classifier, using a Gaussian mixture model, and feeds the likelihood to a classifier. Katzir and Elovici [2018] models the changes in the labels of a K-NN for each activation layer in the base model. Pang *et al.* [2018] propose using a new loss in training, which encourages the neural network to learn latent representations that better distinguish adversarial examples from normal ones. Roth *et al.* [2019] models the statistical robustness of log-odds to perturbations, for normal and adversarial examples. Generally speaking, these methods assume that adversarial examples differ intrinsically from natural images, either in the sample space or because the perturbation affects the propagation of activity in the neural network. Some of those methods require modifying the base model. Very recently, [Fidel *et al.*, 2019] described an explanation-based approach to detection, related to the current work.

2.4 Attacking explainability models

The next step after attacking a model is to attack the explainability model. Such an attack can have multiple desirable affects and can be achieved in several ways. Slack *et al.* [2019] showed a technique that effectively hides the biases of any given classifier, in such a way that its predictions on the input data distribution still remain biased, but the post-hoc explanations of the scaffolded classifier look innocuous. Note that this attack changes the classifier itself, and not a specific example, as is commonly the case in adversarial attacks. Ghorbani *et al.* [2019] showed that two perceptively indistinguishable inputs with the same predicted label can be assigned very different interpretations. Subramanya *et al.* [2019] and Zhang *et al.* [2018b] present a new class of attacks that generate adversarial examples not only misleading the base models but also deceiving their coupled interpretation models. These results indicate that existing interpretability methods are not reliable in the sense that they do not really explain the reason for the model's classification.

Although explainability methods can be fooled, it can be done only when the attacker has full knowledge about the explainability model. In this work, we test the detection models against oblivious adversaries, an attack scenario in which an attacker has full knowledge about the base model (white box attack), but is not aware of the existence of the defense model. We justify this scenario in chapter 4.

Chapter 3

EXAID: EXplAIn then Detect

EXAID consists of two components: (1) **Explain**. Create per-class explanations for both correct predictions and incorrect ones. (2) **Detect**. Train a binary classifier to decide if an explanation is consistent with the class decision. These two components are schematically shown in Figure 1.1.

3.1 Explain

The first step in EXAID implements an explanation model. Given a pretrained classifier that may be attacked, we used an explainability model to extract explanations for every sample classified by the model. The explanation model can take as input the raw input image, as well as the whole base model architecture and weights, and produce an explanation in the terms of the input features. Formally it is a function that maps a sample and a classifier, and its prediction into explanation space $\mathcal{E} : (x, f_{theta}(x)) \to R^n$, where f_{theta} is a classification model producing a predicted label $y = output(f(x))$.

Since our goal is to learn which explanations are typical for each class, we collected both *positive explanations* - applying an explanation model to a correct prediction of the network, and *negative explanations* - corresponding to incorrect predictions of the model.

3.2 Sampling explanations

Creating positive explanations $\mathcal{E}(x_i, y_i)$ is usually straight forward, as one simply applies the explanation model on each sample that was correctly classified $f(x_i) = y_i$. More care should be given to creating negative explanations. We consider three types of negative explanations: ***wrong negatives***, ***adversarial negatives*** and ***other-class negatives***.

First, one may collect samples (x_i, y_i) where the model made an incorrect decision $f(x_i) \neq y_i$, and collect their explanations $\mathcal{E}(x_i, f(y_i))$. We name these ***wrong negatives***. For models that are well trained, the number of these explanations is small. Furthermore, not all classes are confused by other classes, and only some classes may lead to explanations of some other classes.

Second, one can employ an adversarial attack on the training data and collect negative explanations of adversarially perturbed samples. We name them ***adversarial negatives***. As with wrong negatives, these explanations correspond to cases where the model made an incorrect decision, but unlike wrong negatives the explanations may have a different distribution because the input was designed to confuse the network. Even if the specific type of adversarial attack is not known, these samples may be useful because they are based on fooled decisions and may reflect typical patterns of adversarial examples. However, training against an incorrect attack may cause overfitting to a specific type of attack and hurt detection accuracy.

Third, for every labeled sample (x_i, y_i), we produce explanations $\mathcal{E}(x_i, y)$ for all incorrect classes $y \in \mathcal{Y}, y \neq y_i$. For example, for a car image correctly classified as a car, we produce explanations for classes like dogs and cats. These explanations are used as ***other-class negatives*** for the correct class y_i.

3.3 The explainability model

As an explainable AI approach, we used SHAP deep explainer. As shown in [Lundberg and Lee, 2017a] SHAP is considered a leading explainer, providing explanations that have a stronger agreement with human explanations than other methods. We, therefore, believe it is likely to capture the ”correct” features by which people make labeling decisions. In addition, Lundberg and Lee [2017a] has shown that SHAP is the only explainer that has both local accuracy and consistency, which are desirable properties.

3.4 Detect

Given a set of positive and negative explanations per class, we train a deep binary classifier per class, to detect explanations that are inconsistent with model predictions. Note that in this setting, it is natural to train a detector as a binary multiclass multi-label classifier, and not as a multiclass classifier, because we wish to condition the decision on the prediction of the image classifier.

When training the detector, one may consider two learning setups, aiming to protect against *unknown-attacks*, or against *familiar attacks*. It appears as if defending against a known attack would be an easier task, because one may learn the properties of the attack. Unfortunately, since new attacks can be easily designed, it is highly desirable to devise generic defenses.

We address this topic by controlling the data that is used for training the detector. Specifically, we consider two variants of EXAID.

EXAID familiar. During training, the binary detector is presented with adversarial negatives. It can, therefore, learn a distribution of explanations resulting from a specific adversarial attack. Specifically, we trained using high-noise FGSM.

EXAID unknown. The binary detector is not presented with any adversarial negatives during training. The only negative explanation the classifier trained on are other-class negatives and wrong negatives.

Below we tested both variants on the known attack (FGSM) and on unfamiliar attacks (PGD, C&W).

Chapter 4

Experiments

We evaluated EXAID on two benchmark data sets, in the task of attack detection. Our code is available at https://github.com/amosy3/EXAID.

4.1 Datasets

We evaluated EXAID on two standard benchmarks: CIFAR10 [Krizhevsky *et al.*] and SVHN [Netzer *et al.*, 2011]. As Carlini and Wagner [2017a] showed, MNIST is not a good dataset for evaluating adversarial defenses. This is probably due to the fact that it is a low-dimension dataset, making it easier to detect changes an attacker made to the image. [Carlini and Wagner, 2017a] show their results on CIFAR-10. In order to show the validity of our results on more than one dataset, we also used SVHN that has similar complexity. The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. The SVHN dataset is obtained from house numbers in Google Street View images. It consists of more than 600,000 32x32 color images in 10 classes. While similar in flavor to MNIST, it comes from a significantly more diverse distribution. We used the 73,257 digits provided for training and the 26,032 digits for testing.

4.2 Implementation details

For both CIFAR-10 and SVHN we used a pretrained Resnet34 as a base model. To train the EXAID detector we extract positive explanation, wrong negative and other-class negative from natural images as described in algorithm 1. The *EXAID-unknown* model was trained on those explanations. To train *EXAID-familiar* we extracted adversarial negative using a FGSM attack with a noise level of $\epsilon = 0.1$.

Algorithm 1 Create positive and negative explanation, and train bad explanation detector for each class

```
Input: $F$ - Trained model, $(X, Y)$ - Dataset of labeled samples
Initialize: positives ← array of empty sets, negatives ← array of empty sets
for $(x, y) \in (X, Y)$ do
    $\hat{y} = F(x)$
    $explanation = SHAP(F, x)$
    if $\hat{y} == y$ then
        $positives[\hat{y}].append(explanation[\hat{y}])$
        for $i = 1..\hat{y} - 1, \hat{y} + 1..n$ do
            $negatives[i].append(explanation[i])$ ;      // Collect other-class negative
        end for
    else
        $negatives[\hat{y}].append(explanation[\hat{y}])$ ;  // Collect wrong negatives
    end if
end for
for $i = 1..n$ do
    $C_i$ ← Train(positives[i],negatives[i]) ; // Train i-th class classifier
end for
```

As described, we used SHAP as an explainability model [Lundberg and Lee, 2017b]. The original implementation of SHAP runs on CPU and is therefore suitable for use only on a small number of samples, rather than on entire datasets. To enable us to run SHAP on a large number of examples we modified the original implementation to run on GPU. The new version is available in our Git repository.

4.3 Defense Baselines

We compared EXAID with three recently-published adversarial detection baselines, and two new variants of these baselines.

(1) ANR [Liang *et al.*, 2018]. A method based on measuring the effect of quantization and smoothing of the image on the network classification, both are parameterized as a function of the image entropy. We used the implementation provided by the authors. Since ANR was not tested in the original paper on CIFAR-10 and SVHN as done here, we tuned the hyperparameters of their method using hyperopt [Bergstra *et al.*, 2013].

(2) Mahalanobis [Lee *et al.*, 2018]. This approach models the distribution of activations in the hidden layers of the classifier, as obtained in response to natural (unperturbed) samples, using a Gaussian mixture model. Given a set of likelihood scores from the GMM, a classifier is trained to determine if a set of activations is obtained in response to an adversarial example or a natural one. That classifier is trained on adversarial examples. We used the implementation provided by the authors, and as the original paper, we trained the classifier with adversarial examples crafted by FGSM.

(3) Mahalanobis Unsupervised. We modified the method of [Lee *et al.*, 2018] to reach an attack-agnostic baseline as follows. Instead of training an attack-dependent discriminator on adversarial samples, we estimated the likelihood of a set of network activations as the product of likelihoods of all layers.

(4) LID [Ma *et al.*, 2018]. LID measures the characteristics of the region surrounding a reference example, and give it a likelihood score. This is done separately for each representation of the example, in the classifier's hidden layers. As in Mahalanobis, a classifier is trained to determine if a set of activations is obtained in response to an adversarial example or a natural one. We used the implementation from [Lee *et al.*, 2018], and trained the classifier with adversarial examples crafted by FGSM.

(5) Unsupervised LID. As for Mahalnobis, we test an unsupervised version of LID, based on the product of likelihoods of individual layers, without training a classifier.

4.4 Experimental Setup

We test the detection models against *oblivious adversaries*, an attack scenario in which an attacker has full knowledge about the model (white box attack), but is not aware of the existence of the defense model. We believe this is a relevant scenario since, in the real world, most attackers will not have direct access to the attacked model and its defense. In this case, the attacker will be forced to use a black box attack. However, as [Papernot *et al.*, 2017] showed, adversarial examples are transferable between models. Given transferability, attacking a black box model is not marginally harder than a white box. Because of that, we baseline our model against a white box attack as which is a more challenging task.

This is not the case when the model is defended, since [Li *et al.*, 2019] shows that the transferability of adversarial examples works well between vanilla neural networks, but fail to transfer between defended neural networks.

We believe that the magnitude of perturbation used by an attack is a major factor that determines the success of adversarial detection methods. There is still no clear protocol in the literature about comparing attacks and detections depending on this factor, and different reported experiments use different values. We, therefore, repeated all experiments for a wide range of noise levels and report performance across that wide range.

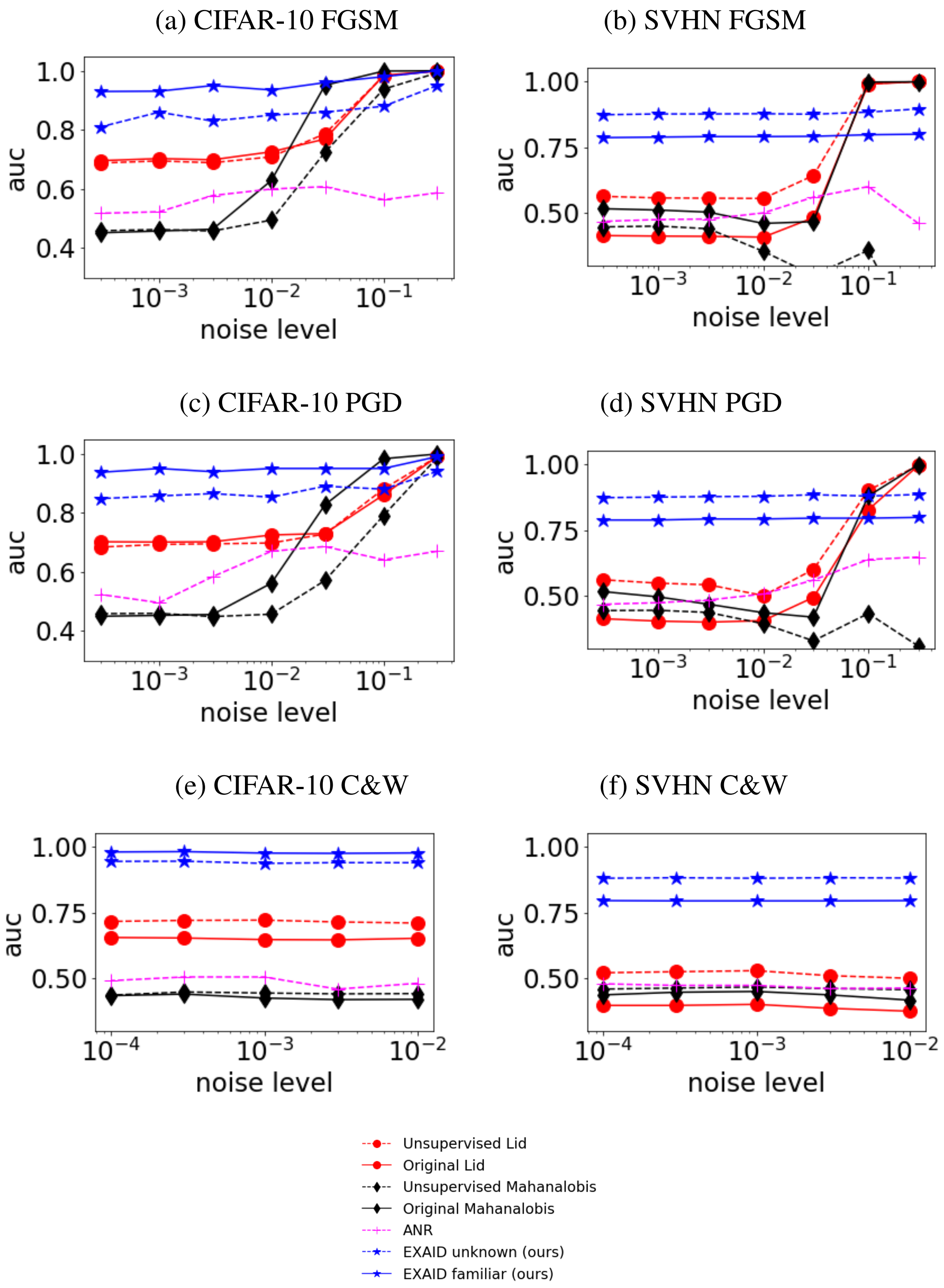
4.5 Attacks

We used three attack methods to test EXAID: (1) One step gradient attack (FGSM) [Goodfellow *et al.*, 2014], (2) Iterative projected gradient (PGD) [Madry *et al.*, 2017] and the Carlini and Wagner attack, which uses optimization to add as small as possible perturbation (C&W) [Carlini and Wagner, 2017b]. All attacks were implemented using Advertorch [Ding *et al.*, 2019]. Opposed to other defense methods benchmarks, we examined the effect of noise-level on a range of three orders of magnitude.

4.6 Results

The results for all detection methods are shown in figure 4.1. EXAID significantly outperforms the other methods when the noise level is small (small perturbations), and with attack methods that use adaptive noise levels (C&W). Typically, the AUC is increased from 70% to over 90%.

LID and Mahalanobis both perform well in high noise scenarios, and slightly outperform EXAID on SVHN in these scenarios. However, when the noise level decreases LID and Mahalanobis performance suffers drastically, while EXAID's remains high. Interestingly, our unsupervised variant of LID, performs at least as well, and sometimes better, than the original LID. This may be because LID was trained with FGSM samples and may deteriorate in cross-attack scenarios. These findings show the importance of benchmarking defense models against a wide range of noise levels.



(a) CIFAR-10 FGSM

(b) SVHN FGSM

(c) CIFAR-10 PGD

(d) SVHN PGD

(e) CIFAR-10 C&W

(f) SVHN C&W

Figure 4.1: **Defense methods comparison**. Each sub-figure compares 2 EXAID variants to five baselines. (a,b) defend against FGSM, for CIFAR-10 and SVHN. (c,d) same against PGD (d,e) Same against C&W. EXAID outperforms all baselines in low-noise scenarios, and is comparable in the high-noise regime.

Chapter 5

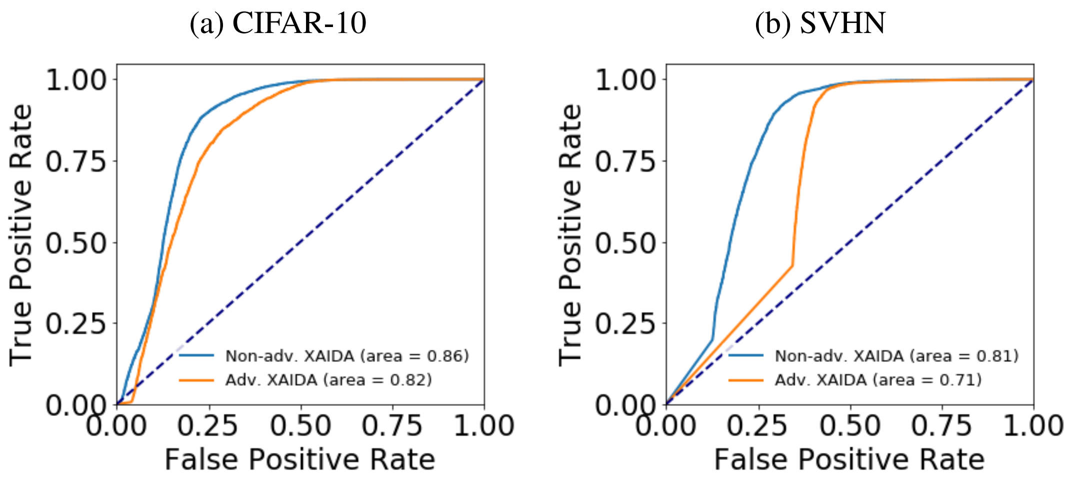
Detecting out-of-distribution samples

While the main purpose of this work was to detect adversarial examples, it also has implications in detecting abnormal samples that are drawn far away from the distribution of training samples. In a sense, all the statistical methods for adversarial detection assume adversarial examples don't come from the same distribution as natural examples. Detecting out-of-distribution samples is an important task, as it was shown neural networks don't generalize well across different distributions Liang *et al.* [2017]. However, when deploying a neural network in real-world applications, there is little control over the input data's distribution. Recent works have also shown that neural networks tend to make high confidence predictions even for completely irrelevant inputs [Liang *et al.*, 2017]. Therefore, being able to accurately detect out-of-distribution examples can be practically important.

EXAID uses a negative explanation detector to find adversarial examples. We tested its ability to detect out-of-distribution examples, under the assumption that the explanations of those out-of-distribution examples are in the distribution of the detector. We have also assumed the same for adversarial examples, but for out-of-distribution cases, it is a stronger assumption: Even if we look on adversarial examples as out-of-distribution ones, they are crafted in a way that lets them stay very close to the original distribution. In contrast, in the out-of-distribution task, a detector should detect out-of-distribution examples even when they are far away from the distribution of training samples.

For this experiment, we used the same detector from chapter 3, with no additional tuning for the new task. As in Liang *et al.* [2017] and Lee *et al.* [2018] we check the ability of the detector to detect examples from another dataset as out-of-distribution. In order to do that, we take a pretrained model on CIFAR-10, and ran it on examples from SVHN. We used SHAP to explain the predicted class, and used both versions of EXAID detector from chapter 3 to classify the explanation as positive or negative. This time, a negative explained example will be declared as an out-of-distribution (and not adversarial) example. We also repeated the experiment in the opposite direction. The results are in figure 5.1.

According to Lee *et al.* [2018] state of the art out-of-distribution detectors like Liang *et al.* [2017] gets an AUC score of 0.96 for detecting SVHN examples from CIFAR-10 distribution, and an AUC score of 0.91 for detecting CIFAR-10 examples from SVHN distribution. Although our AUC scores of 0.86 and 0.81 respectively, are not as good as the state of the art, we believe they may help to improve other out-of-distribution methods, due to the fact that our method uses different properties of the example.



(a) CIFAR-10

(b) SVHN

Figure 5.1: **EXAID use for out-of-distribution task.** ROC curve for both EXAID versions from chapter 3 at out-of-distribution detection. (a) SVHN examples detection from CIFAR-10 distribution. (b) CIFAR-10 examples detection from SVHN distribution.

Chapter 6

Conclusion

In this work, we proposed EXAID, a novel attack-detection approach, which uses model explainability to identify images whose explanations are inconsistent with the predicted class. Our method outperforms previous state-of-the-art methods, for three attack methods, and many noise-levels. We demonstrated that the attack noise level has a major impact on previous defense methods. We hope this will encourage the research community to evaluate future defense methods on a large range of noise-levels.

Bibliography

James Bergstra, Dan Yamins, and David D Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, pages 13–20. Citeseer, 2013.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, pages 8928–8939, 2019.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976, 2017.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

Gil Fidel, Ron Bitton, and Asaf Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. *arXiv preprint arXiv:1909.03418*, 2019.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

Ziv Katzir and Yuval Elovici. Detecting adversarial perturbations through spatial behavior in activation spaces. *arXiv preprint arXiv:1811.09043*, 2018.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *arXiv preprint arXiv:1905.00441*, 2019.

Shiyu Liang, Yixuan Li, and R. Srikant. Principled detection of out-of-distribution examples in neural networks. 06 2017.

Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi, and XiaoFeng Wang. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. *IEEE Transactions on Dependable and Secure Computing*, 2018.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Margarita Osadchy, Julio Hernandez-Castro, Stuart Gibson, Orr Dunkelman, and Daniel Pérez-Cabo. No bot expects the deepcaptcha! introducing immutable adversarial examples, with applications to captcha generation. *IEEE Transactions on Information Forensics and Security*, 12(11):2640–2653, 2017.

Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. Towards robust detection of adversarial examples. In *Advances in Neural Information Processing Systems*, pages 4579–4589, 2018.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017.

Mikhail Pautov, Grigorii Melnikov, Edgar Kaziakhmedov, Klim Kireev, and Aleksandr Petiushko. On adversarial patches: real-world attack on arcface-100 face recognition system. *arXiv preprint arXiv:1910.07067*, 2019.

Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8571–8580, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. *arXiv preprint arXiv:1902.04818*, 2019.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. How can we fool lime and shap? adversarial attacks on post hoc explanation methods. *arXiv preprint arXiv:1911.02508*, 2019.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019.

Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2029, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.

Xinyang Zhang, Ningfei Wang, Shouling Ji, Hua Shen, and Ting Wang. Interpretable deep learning under fire. *arXiv preprint arXiv:1812.00891*, 2018.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

תקציר

מודלים של למידה עמוקה רגישים לדוגמאות משטות. דוגמאות משטות הן דוגמאות שעוצבו באופן מדויק על מנת לגרום למערכת לבצע החלטה שגויה. בעבודה זו אנו התמקדנו בזיהוי דוגמאות כאלה, ולא ביצירת מערכות חסינות לדוגמאות מסוג זה, לאור העובדה שזיהוי תקיפה פעמים רבות חשוב הרבה יותר מאשר ההימנעות מביצוע שגיאה. לשם כך השתמשנו במודלי הסבר, המציינים את התרומה של כל פיקסל בקלט, לפלט הסופי שהתקבל. האבחנה המרכזית מבוססת על כך שתוקף יתקשה לתת הסבר משכנע לתמונה שסווגה באופן שגוי – שכן הסבר לא נכון לא אמור לקבל תמיכה מהקלט.

אנו גישה חדשה לזיהוי תקיפות, שמשתמשת במודלי הסבר כדי לזהות הסברים שאינם תואמים את החיזוי שנתנה המערכת. השתמשנו ב-SHAP, שיטה המקרבת את ערכי שאפלי עבור התמונה המתקבלת בקלט כדי לזהות את התרומה של כל פיקסל להחלטה של המערכת. שיטה זו איננה דורשת שינוי כלשהו במודל הנתקף ויכולה להיות מיושמת ללא צורך בידע מוקדם על סוג התקיפה ממנו נדרש להתגונן. בשל כך, שיטה זו יכולה לעזור בהתמודדות עם תקיפות שלא היו ידועות בזמן תכנון המודל בכלל ומודל ההגנה בפרט.

השיטה נבחנה אל מול שני בסיסי נתונים מקובלים - CIFAR-10 ו-SVHN ואל מול שלוש שיטות תקיפה מובילות FGSM, PGD ו-C&W . אנו הראנו שהשיטה המוצעת משפרת את השיטות המובילות בתחום בהפרש ניכר בטווח רחב של ערכי רעש שהתוקף מורשה להכניס לתמונה, כאשר עבור ערכי רעש קטנים יכולת הגילוי משתפרת מ-70% ל-90%.

עבודה זו נעשתה בהדרכתם של

פרופסור גל צ'צ'יק ופרופסור אלי פורת

מן המרכז הרב תחומי לחקר המוח והמחלקה למדעי המחשב של אוניברסיטת בר אילן

אוניברסיטת בר-אילן

שימוש במודלי הסבר לשם זיהוי דוגמאות משטות

אוהד עמוסי

עבודה שוות ערך לתזה מוגשת כחלק מדרישות קבלה ללימודים לתואר שלישי במחלקה למדעי המחשב של אוניברסיטת בר-אילן

רמת גן תשע"ט