SHIELD: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression

Nilaksh Das[1], Madhuri Shanbhogue[1], Shang-Tse Chen[1], Fred Hohman[1], Siwei Li[1], Li Chen[2], Michael E. Kounavis[2], Duen Horng Chau[1]
[1]Georgia Institute of Technology, Atlanta, GA, USA
{nilakshdas,madhuri.shanbhogue,schen351,fredhohman,robertsiweili,polo}@gatech.edu
[2]Intel Corporation, Hillsboro, OR, USA
{li.chen,michael.e.kounavis}@intel.com

ABSTRACT

The rapidly growing body of research in adversarial machine learning has demonstrated that deep neural networks (DNNs) are highly vulnerable to adversarially generated images. This underscores the urgent need for practical defense techniques that can be readily deployed to combat attacks in real-time. Observing that many attack strategies aim to perturb image pixels in ways that are visually imperceptible, we place JPEG compression at the core of our proposed SHIELD defense framework, utilizing its capability to effectively "compress away" such pixel manipulation. To immunize a DNN model from artifacts introduced by compression, SHIELD "vaccinates" the model by retraining it with compressed images, where different compression levels are applied to generate multiple vaccinated models that are ultimately used together in an ensemble defense. On top of that, SHIELD adds an additional layer of protection by employing randomization at test time that compresses different regions of an image using random compression levels, making it harder for an adversary to estimate the transformation performed. This novel combination of vaccination, ensembling, and randomization makes SHIELD a fortified multi-pronged defense. We conducted extensive, large-scale experiments using the ImageNet dataset, and show that our approaches eliminate up to 98% of gray-box attacks delivered by strong adversarial techniques such as *Carlini-Wagner's L2* attack and *DeepFool*. Our approaches are fast and work without requiring knowledge about the model.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Machine learning**; **Neural networks**; **Image compression**;

KEYWORDS

adversarial machine learning; JPEG compression; deep learning; machine learning security; ensemble defense

KDD 2018, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219910

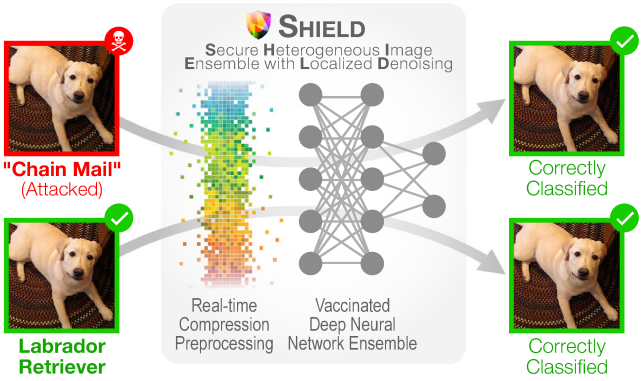

Figure 1: SHIELD Framework Overview. SHIELD combats adversarial images (in red) by removing perturbation in real-time using Stochastic Local Quantization (SLQ) and an ensemble of vaccinated models which are robust to the compression transformation. Our approach eliminates up to 98% of gray-box attacks delivered by strong adversarial techniques such as *Carlini-Wagner's L2* attack and *DeepFool*.

ACM Reference Format:
Nilaksh Das[1], Madhuri Shanbhogue[1], Shang-Tse Chen[1], Fred Hohman[1], Siwei Li[1], Li Chen[2], Michael E. Kounavis[2], Duen Horng Chau[1]. 2018. SHIELD: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression . In *KDD 2018: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3219819.3219910

1 INTRODUCTION

Deep neural networks (DNNs), while enjoying tremendous success in recent years, suffer from serious vulnerabilities to adversarial attacks [33]. For example, in computer vision applications, an attacker can add visually imperceptible perturbations to an image and mislead a DNN model into making arbitrary predictions. When the attacker has complete knowledge of a DNN model, these perturbations can be computed by using the gradient information of the model, which guides the adversary in discovering vulnerable regions of the input space that would most drastically affect the model output [11, 27]. But even in a black-box scenario, where the attacker does not know the exact network architecture, one can use a substitute model to craft adversarial perturbations that are transferable to the target model [25]. To make this even more

troubling, it is possible to print out physical 2D or 3D objects to fool recognition systems in realistic settings [2, 30].

The threat of adversarial attack casts a shadow over deploying DNNs in security and safety-critical applications like self-driving cars. To better understand and fix the vulnerabilities, there is a growing body of research on defending against various attacks and making DNN models more robust [3, 22, 26]. However, the progress of defense research has been lagging behind the attack side so far. Moreover, research on defense rarely focuses on practicality and scalability, both essential for real-world deployment. For example, total variation denoising and image quilting are image preprocessing techniques that have potential in mitigating adversarial perturbations to some extent [14], but they incur significant computational overhead, calling into question how feasibly they can be used in practical applications, which often require fast, real-time defense [8, 9].

1.1 Our Contributions and Impact

1. Compression as Fast, Practical, Effective Defense. We leverage the idea that *compression* — a central concept that underpins numerous successful data mining techniques — can offer powerful, scalable, and practical protection for deep learning models against adversarial image perturbations in real-time. Motivated by the observation that many attack strategies aim to perturb images in ways that are visually imperceptible to the naked eye, we show that systematic adaptation of the widely available JPEG compression technique can effectively compress away such pixel "noise", especially since JPEG is particularly designed to reducing image details that are imperceptible to humans. (Section 3.1)

2. SHIELD: Multifaceted Defense Framework. Building on our principal idea of compression, we contribute the novel SHIELD defense framework that combines *randomization*, *vaccination* and *ensembling* into a fortified multi-pronged defense:

(1) We exploit JPEG's flexibility in supporting varying compression levels to develop strong ensemble models that span the spectrum of compression levels;
(2) We show that a model can be "vaccinated" by training on compressed images, increasing its robustness towards compression transformation for both adversarial and benign images;
(3) SHIELD employs stochastic quantization that compresses different regions of an image using randomly sampled compression levels, making it harder for the adversary to estimate the transformation performed.

SHIELD does not require any change in the model architecture, and can recovers significant amount of model accuracy lost to adversarial instances, with little effect on the accuracy for benign instances. SHIELD stands for **S**ecure **H**eterogeneous **I**mage **E**nsemble with **L**ocalized **D**enoising. (Sections 3.2 & 3.3)

3. Extensive Evaluation Against Major Attacks. We perform extensive experiments using the full ImageNet benchmark dataset with 50K images, demonstrating that our approach is fast, effective and scalable. Our approaches eliminate up to 98% of gray-box attacks delivered by some of the most recent, strongest attacks, such as *Carlini-Wagner's L2* attack [4] and *DeepFool* [24]. (Section 4)

4. Impact to Intel and Beyond. This work is making multiple positive impacts on Intel's research and product development plans. Introduced with the Sandy Bridge CPU microarchitecture, Intel's Quick Sync Video (QSV) technology dedicates a hardware core for high-speed video processing, performs JPEG compression up to 24X faster than TensorFlow implementations, paving the way for real-time defense in safety-critical applications, such as autonomous vehicles. This research has sparked insightful discussion among research and development teams at Intel, on the priority of *secure deep learning* that necessitates tight integration of practical defense strategies, software platforms and hardware accelerators. We believe our work will accelerate the industry's emphasis on this important topic. To ensure reproducibility of our results, we have open-sourced our code on GitHub (https://github.com/poloclub/jpeg-defense). (Section 5)

2 BACKGROUND: ADVERSARIAL ATTACKS

Our work focuses on defending against adversarial attacks on deep learning models. This section provides background information for readers new to the adversarial attack literature.

Given a trained classifier C and an instance $x \in X$, the objective of an adversarial *untargeted* attack is to compute a perturbed instance x' such that $C(x') \neq C(x)$ and $d(x, x') \leq \rho$ for some distance function $d(\cdot, \cdot)$ and $\rho \geq 0$. Popular choices of $d(\cdot, \cdot)$ are Euclidean distance $d(x, x') = \|x - x'\|_2$, and Chebychev distance $d(x, x') = \|x - x'\|_\infty$. A *targeted* attack is similar, but is required to induce a classification for a specific target class t, i.e., $C(x') = t$. In both cases, depending on whether the attacker has full knowledge of C or not, the attack can be further categorized into *white-box* attack and *black-box* attack. The latter is obviously harder for the attacker since less information is known about the model, but has been shown to be possible in practice by relying on the property of transferability from a substitute model to the target model when both of them are DNNs trained using gradient backpropagation [25, 33].

The seminal work by Szegedy et al. [33] proposed the first effective adversarial attack on DNN image classifiers by solving a box-constrained L-BFGS optimization problem and showed that the computed perturbations to the images were indistinguishable to the human eye — a rather troublesome property for people trying to identify adversarial images. This discovery has gained tremendous interest, and many new attack algorithms have been invented [11, 23, 24, 27] and applied to other domains such as malware detection [12, 15], sentiment analysis [28], and reinforcement learning [16, 20]. Below, we describe the major, well-studied attacks in the literature, against which we will evaluate our approach.

Carlini-Wagner's L_2 (*CW-L2*) [4] is an optimization-based attack that adds a relaxation term to the perturbation minimization problem based on a differentiable surrogate of the model. They pose the optimization as minimizing:

$$\|x - x'\|_2 + \lambda \max\big(-\kappa, Z(x')_k - \max\{Z(x')_{k'} : k' \neq k\}\big) \quad (1)$$

where κ controls the confidence with which an image is misclassified by the DNN, and $Z(\cdot)$ is the output from the logit layer (last layer before the softmax function is applied for prediction) of C.

DeepFool (*DF*) [24] constructs an adversarial instance under an L_2 constraint by assuming the decision boundary to be hyperplanar. The authors leverage this simplification to compute a minimal adversarial perturbation that results in a sample that is close to the original instance but orthogonally cuts across the nearest decision boundary. In this respect, *DF* is an untargeted attack. Since the underlying assumption about the decision boundary being completely linear in higher dimensions is an oversimplification of the actual case, *DF* keeps reiterating until a true adversarial instance is found. The resulting perturbations are harder for humans to detect compared to perturbations introduced by other attacks.

Iterative Fast Gradient Sign Method (*I-FGSM*) [19] is the iterative version of the **Fast Gradient Sign Method** (*FGSM*) [11], which is a fast algorithm that computes perturbations subject to an L_∞ constraint. *FGSM* simply takes the sign of the gradient of loss function J w.r.t. the input x,

$$x' = x + \epsilon \cdot sign(\nabla J_x(\theta, x, y)) \tag{2}$$

where θ is the set of parameters of the model and y is the true label of the instance. The parameter ϵ controls the magnitude of per-pixel perturbation. *I-FGSM* iteratively applies FGSM in each iteration i after clipping the values appropriately at each step:

$$x^{(i)} = x^{(i-1)} + \epsilon \cdot sign(\nabla J_{x^{(i-1)}}(\theta, x^{(i-1)}, y)) \tag{3}$$

3 PROPOSED METHOD: COMPRESSION AS DEFENSE

In this section, we present our compression-based approach for combating adversarial attacks. In Section 3.1, we begin by describing the technical reasons why compression can remove perturbation. As compression would modify the distribution of the input space by introducing some artifacts, in Section 3.2, we propose to "vaccinate" the model by training it with compressed images, which increases its robustness towards compression transformation for both adversarial and benign images. Finally, in Section 3.3, we present our multifaceted SHIELD defense framework that combines random quantization, vaccination and ensembling into a fortified multi-pronged defense, which, to the best of our knowledge, has yet been challenged.

3.1 Preprocessing Images using Compression

Our main idea on rectifying the prediction of a trained model C, with respect to a perturbed input x', is to apply a preprocessing operation $g(\cdot)$ that brings back x' closer to the original benign instance x, which implicitly aims to make $C(g(x')) = C(x)$. Constructing such a $g(\cdot)$ is application dependent. For the image classification problem, we show that JPEG compression is a powerful preprocessing defense technique. JPEG compression mainly consists of the following steps:

(1) Convert the given image from *RGB* to YC_bC_r (chrominance + luminance) color space.
(2) Perform spatial subsampling of the chrominance channels, since the human eye is less susceptible to these changes and relies more on the luminance information.
(3) Transform 8×8 blocks of the YC_bC_r channels to a frequency domain representation using Discrete Cosine Transform (DCT).

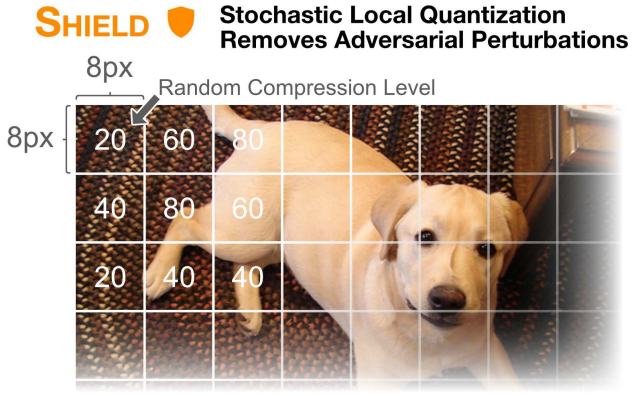

Figure 2: SHIELD uses Stochastic Local Quantization (SLQ) to remove adversarial perturbations from input images. SLQ divides an image into 8×8 blocks and applies a randomly selected JPEG compression quality (20, 40, 60 or 80) to each block to mitigate the attack.

(4) Perform quantization of the blocks in the frequency domain representation according to a quantization table which corresponds to a user-defined quality factor for the image.

The last step is where the JPEG algorithm achieves the majority of compression at the expense of image quality. This step suppresses higher frequencies more since these coefficients contribute less to the human perception of the image. As adversarial attacks do not optimize for maintaining the spectral signature of the image, they tend to introduce more high frequency components which can be removed at this step. This step also renders the preprocessing stage non-differentiable, which makes it non-trivial for an adversary to optimize against, allowing only estimations to be made of the transformation [31]. We show in our evaluation (Section 4.2) that JPEG compression effectively removes adversarial perturbation across a wide range of compression levels.

3.2 Vaccinating Models with Compressed Images

As DNNs are typically trained on high quality images (with little or compression), they are often invariant to the artifacts introduced by the preprocessing of JPEG at high-quality settings. This is especially useful in an adversarial setting as our pilot study has shown that applying even mild compression removes the perturbations introduced by some attacks [6]. However, applying too much compression could reduce the model accuracy on benign images.

We propose to "vaccinate" the model by training it with compressed images, especially those at lower JPEG qualities, which increases the model's robustness towards compression transformation for both adversarial and benign images. With vaccination, we can apply more aggressive compression to remove more adversarial perturbation. In our evaluation (Section 4.3), we show the significant advantage that our vaccination strategy provides, recovering more than 7 *absolute* percentage points in model accuracy for high-perturbation attacks.

3.3 SHIELD: Multifaceted Defense Framework

To leverage the effectiveness of JPEG compression as a preprocessing technique along with the benefit of vaccinating with JPEG images, we propose a *stochastic variant* of the JPEG algorithm that introduces randomization to the quantization step, making it harder for the adversaries to estimate the preprocessing transformation.

Figure 2 illustrates our proposed strategy, where we vary the quantization table for each 8×8 block in the frequency domain to correspond to a random quality factor from a provided set of qualities, such that the compression level does not remain uniform across the image. This is equivalent to breaking up the image into disjoint 8×8 blocks, compressing each block with a random quality factor, and putting the blocks together to re-create the final image. We call this method *Stochastic Local Quantization* (SLQ). As the adversary is free to craft images with varying amounts of perturbation, our defense should offer protection across a wide spectrum. Thus, we selected the set of qualities $\{20, 40, 60, 80\}$ as our randomization candidates, uniformly spanning the range of JPEG qualities from 1 (most compressed) to 100 (least compressed).

Comparing our stochastic approach to taking a simple average over JPEG compressed images, our method allows for maintaining the original semantics of the image in the blocks compressed to higher qualities, while performing more localized denoising in the blocks compressed to lower qualities. In the case of simple average, all perturbations may not be removed at higher qualities and they might simply dominate the other components participating in the average, still posing to be adversarial. Introducing localized stochasticity reduces this expectation.

In our evaluation (Section 4.3), we will show that by using the spectrum of JPEG compression levels with our stochastic approach, our model can simultaneously attain a high accuracy on benign images, while being more robust to adversarial perturbations — a strong benefit that using a single JPEG quality cannot provide. Our method is further fortified by using an ensemble of vaccinated models individually trained on the set of qualities picked for randomization. We show in Section 4.3 how our method can achieve high model accuracies, comparable to those of much larger ensembles, but is significantly faster.

4 EVALUATION

In this section, we show that our approach is scalable, effective and practical in removing adversarial image perturbations. For our experiments, we consider the following scenarios:

- The adversary has access to the full model, including its architecture and parameters. (Section 4.2)
- The adversary has access to the model architecture, but not the exact parameters. (Section 4.3)
- The adversary does not have access to the model architecture. (Section 4.4)

4.1 Experiment Setup

We performed experiments on the full validation set of the *ImageNet* benchmark image classification dataset [17], which consists of 1,000 classes, totaling 50,000 images. We show the performance of each defense on the *ResNet-v2 50* model obtained from the *TF-Slim* module in *TensorFlow*. We construct the attacks using the popular

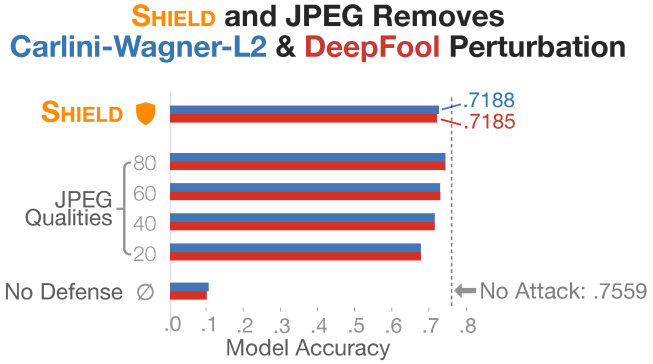

Figure 3: Carlini-Wagner-L2 (CW-L2) and DeepFool, two recent strong attacks, introduce perturbations that lowers model accuracy to around 10% (∅). JPEG compression recovers up to 98% of the original accuracy (with DeepFool), while SHIELD achieves similar performance, recovering up to 95% of the original accuracy (with DeepFool).

CleverHans package[1], which contains implementations from the authors of the attacks.

- For *Carlini-Wagner-L2* (CW-L2), we set its parameter $\kappa = 0$, a common value used in studies [14], as larger values (higher confidence) incur prohibitively high computation cost.
- *DeepFool* (DF) is a non-parametric attack that optimizes the amount of perturbation required to misclassify an image.
- For *FGSM* and *I-FGSM*, we vary ϵ from 0 to 8 in steps of 2.

We compare JPEG compression and SHIELD with two popular denoising techniques that have potential in defending against adversarial attacks [14, 35]. Median filter (MF) collapses a small window of pixels into a single value, and may drop some of the adversarial pixels in the process. Total variation denoising (TVD) aims to reduce the total variation in an image, and may undo the artificial noise injected by the attacks. We vary the parameters of each method to evaluate how their values affect defense performance.

- For JPEG compression, we vary the compression level from quality 100 (least compressed) to 20 (greatly compressed), in decrements of 10.
- For *median filter* (MF), we use window sizes of 3 (smallest possible) and 5. We tested larger window sizes (e.g., 7), which led to extremely poor model accuracies, thus were ruled out as parameter candidates.
- For *total variation denoising* (TVD), we vary its weight parameter from 10 through 40, in increments of 10. Reducing the weight of TVD further (e.g., 0.3) produces blurry images that lead to poor model accuracy.

4.2 Defending Gray-Box Attacks with Image Preprocessing

In this section, we investigate the setting where an adversary gains access to all parameters and weights of a model that is trained

[1] https://github.com/tensorflow/cleverhans

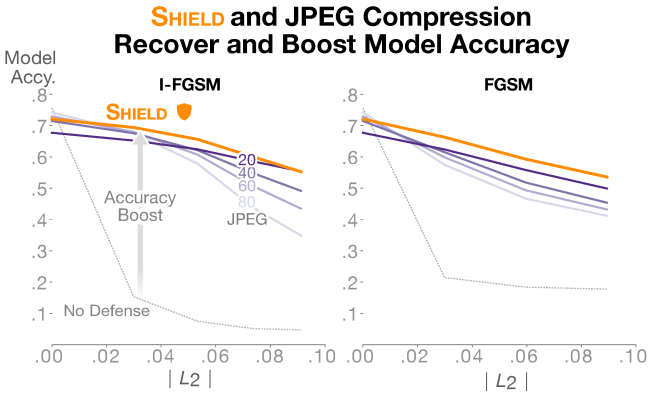

Figure 4: SHIELD recovers the accuracy of the model when attacked with I-FGSM (left) and FGSM (right). Both charts show the accuracy of the model when undefended (gray dotted curve). Applying varying JPEG compression qualities (purple curves) helps recover accuracy significantly, and SHIELD (orange curve) is able to recover more than any single JPEG-defended model.

on benign images, but is unaware of the defense strategy. This constitutes a *gray-box* attack on the overall classification pipeline.

We show the results of applying JPEG compression at various qualities on images attacked with Carlini-Wagner-L2 (CW-L2) and DeepFool (DF) in Figure 3, and on images attacked with I-FGSM and FGSM in Figure 4.

Combating Carlini-Wagner-L2 (CW-L2) & DeepFool (DF). Although CW-L2 and DF, both considered strong attacks, are highly effective at lowering model accuracies, Figure 3 shows that even applying mild JPEG compression (i.e., using higher JPEG qualities) can recover much of the lost accuracy. Since both methods optimize for a lower perturbation to fool the model, the noise introduced by these attacks is imperceptible to the human eye and lies in the high frequency spectrum, which is destroyed in the quantization step of the JPEG algorithm. SHIELD performs well, and comparably, for both attacks. We do not arbitrarily scale the perturbation magnitude of either attack as in [14], as doing so would violate the attacks' optimization criteria.

Combating I-FSGM & FGSM. As shown in Figure 4, JPEG compression also achieves success in countering I-FGSM and FGSM attacks, which introduce higher magnitudes of perturbation.

As the amount of perturbation increases, the accuracies of models without any protection (gray dotted curves in Figure 4) rapidly falls beneath 19%. JPEG recovers significant portions of the lost accuracies (purple curves); its effectiveness also gradually and expectantly declines as perturbation becomes severe. Applying more compression generally recovers more accuracy (e.g., dark purple curve, for JPEG quality 20), but at the cost of losing some accuracy for benign images. SHIELD (orange curve) offers a desirable trade-off, achieving good performance under severe perturbation while retaining accuracies comparable to the original models. Applying less compression (light purple curves) performs well with benign images but is not as effective when perturbation increases.

Effectiveness and Runtime Comparison against Median Filter (MF) and Total Variation Denoising (TVD). We compare JPEG compression and SHIELD with MF and TVD, two popular denoising techniques, because they too have potential in defending against adversarial attacks [14, 35]. Like JPEG, both MF and TVD are parameterized. Table 1 summarizes the performance of all the image preprocessing techniques under consideration. While all techniques are able to recover accuracies from CW-L2 and DF, both strongly optimized attacks with lower perturbation strength, the best performing settings are from JPEG (bold font in Table 1). When faced with large amount of perturbation generated by the I-FGSM and FSGM attacks, SHIELD benefits from the combination of Stochastic Local Quantization, vaccination, and ensembling, outperforming all other techniques.

As developing practical defense is our primary goal, effectiveness, while important, is only one part of our desirable solution. Another critical requirement is that our solution be fast and scalable. Thus, we also compare the runtimes of the image processing techniques. Our comparison focuses on the most computationally intensive parts of each technique, ignoring irrelevant overheads (e.g., disk I/O) common to all techniques. All runtimes are averaged over 3 runs, using the full 50k ImageNet validation images, on a dedicated desktop computer equipped with an Intel i7-4770K quad-core CPU clocked at 3.50GHz, 4x8GB RAM, 1TB SSD of Samsung 840 EVO-Series and 2x3TB WD 7200RPM hard disk, running Ubuntu 14.04.5 LTS and Python 2.7. We used the fastest, most popular Python implementations of the image processing techniques. We used JPEG and MF from Pillow 5.0, and TVD from scikit-image.

As shown in Figure 5, JPEG is the fastest, spending no more than 107 seconds to compress 50k images (at JPEG quality 80). It is at least 22x faster than TVD, and 14x faster than median filter. We tested the speed of the TensorFlow implementation of SHIELD, which also compresses all images at high speed, taking only 150s.

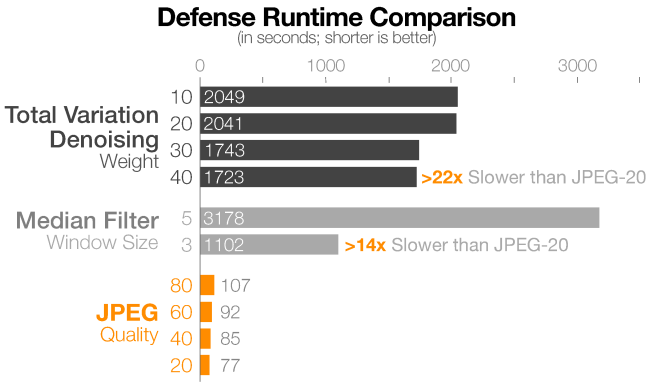

Figure 5: Runtime comparison for three defenses: (1) total variation denoising (TVD), (2) median filter (MF), and (3) JPEG compression, timed using the full 50k ImageNet validation images, averaged over 3 runs. JPEG is at least 22x faster than TVD, and 14x faster than MF.

Defense	No Attack $\lvert L_2\rvert = 0$	**CW-L2** ($\kappa = 0$) $\lvert L_2\rvert = .0025$	**DF** $\lvert L_2\rvert = .0020$	**I-FGSM** ($\epsilon = 4$) $\lvert L_2\rvert = .0533$	**FGSM** ($\epsilon = 4$) $\lvert L_2\rvert = .0597$
No Defense	*75.59*	*10.29*	*9.78*	*7.49*	*18.40*
SHIELD [20, 40, 60, 80]	72.11	71.85	71.88	**65.63**	**59.29**
JPEG [quality=100]	**74.95**	74.37	**74.41**	52.52	44.00
JPEG [quality=90]	74.83	**74.43**	74.36	55.18	45.12
JPEG [quality=80]	74.23	73.92	73.88	57.86	46.66
JPEG [quality=70]	73.61	73.11	73.17	59.53	47.96
JPEG [quality=60]	72.97	72.46	72.52	60.74	49.33
JPEG [quality=50]	72.32	71.86	71.91	61.47	50.53
JPEG [quality=40]	71.48	71.03	71.05	62.14	51.81
JPEG [quality=30]	70.08	69.63	69.67	62.52	53.51
JPEG [quality=20]	67.72	67.32	67.34	62.43	55.81
MF [window=3]	71.05	70.44	70.42	60.09	51.06
MF [window=5]	58.48	58.19	58.06	53.59	49.71
TVD [weight=10]	69.14	68.69	68.74	62.40	53.56
TVD [weight=20]	71.87	71.44	71.45	61.90	50.26
TVD [weight=30]	72.82	72.34	72.37	60.70	48.18
TVD [weight=40]	73.31	72.90	72.91	59.60	47.07

Table 1: Summary of model accuracies (in %) for all defenses: SHIELD, JPEG, median filter (MF), and total variation denoising (TVD); v/s all attacks: Carlini-Wagner L2 (CW-L2), DeepFool (DF), I-FGSM and FGSM. While all techniques are able to recover accuracies from CW-L2 and DF, both strongly optimized attacks with lower perturbation strength, the best performing settings are from JPEG (in bold font). SHIELD benefits from the combination of Stochastic Local Quantization, vaccination and ensembling, outperforming all other techniques when facing high perturbation delivered by I-FGSM and FGSM.

4.3 Black-Box Attack with Vaccination and Ensembling

We now turn our attention to the setting where an adversary has knowledge of the model being used but does not have access to the model parameters or weights. More concretely, we vaccinate the ResNet-v2 50 model by retraining on the ImageNet training set and preprocessing the images with JPEG compression while training. This setup constitutes a *black-box* attack, as the attacker only has access to the original model but not the vaccinated model being used.

We denote the original ResNet-v2 50 model as $\mathcal{M}$, which the adversary has access to. By retraining on images of a particular JPEG compression quality q, we transform $\mathcal{M}$ to $\mathcal{M}_q$, e.g., for JPEG-20 Vaccination, we retrain $\mathcal{M}$ on JPEG-compressed images at quality 20 and obtain $\mathcal{M}_{20}$. When retraining the ResNet-v2 50 models, we used stochastic gradient descent (SGD) with a learning rate of 5×10^{-3}, with a decay of 94% over 25×10^4 iterations. We conducted the retraining on a GPU cluster with 12 NVIDIA Tesla K80 GPUs. In this manner, we obtain 8 models from quality 20 through quality 90 in increments of 10 ($\mathcal{M}_{20}, \mathcal{M}_{30}, \mathcal{M}_{40} \ldots \mathcal{M}_{90}$), to cover a wide spectrum of JPEG qualities. Figure 6 shows the results of model vaccination against FGSM attacks, whose parameter ϵ ranges from 0 (no perturbation) to 8 (severe perturbation), in steps of 2. The plots show that retraining the model helps recover even more model accuracy than using JPEG preprocessing alone (compare the *unvaccinated* gray dotted curve vs. the *vaccinated* orange and purple curves in Figure 6). We found that a given model $\mathcal{M}_q$ performed best when tested with JPEG-compressed images of the same quality q, which was expected.

We test these models in an ensemble with two different voting schemes. The first ensemble scheme, denoted as $\mathcal{M}_q \times q$, corresponds to each model $\mathcal{M}_q$ casting a vote on every JPEG quality q from $q \in \{20, 30, 40, \ldots, 90\}$. This has a total cost of 64 votes, from which we derive the majority vote. In the second scheme, denoted by $\mathcal{M}_q - q$, each model $\mathcal{M}_q$ votes only on q, the JPEG quality it was trained on. This incurs a cost of 8 votes.

Table 2 compares the accuracies (against FGSM) and computation costs of these two schemes with those of SHIELD, which also utilizes an ensemble ($\mathcal{M}_{20}, \mathcal{M}_{40}, \mathcal{M}_{60}, \mathcal{M}_{80}$) with a total of 4 votes. SHIELD achieves very similar performance as compared to the vaccinated models, at half the cost when compared to $\mathcal{M}_q - q$. Hence, SHIELD offers a favorable trade-off in terms of scalability with minimal effect on accuracy.

4.4 Transferability in Black-Box Setting

In this setup, we evaluate the transferability of attacked images generated using ResNet-v2 50 on ResNet-v2 101 and Inception-v4. The attacked images are preprocessed using JPEG compression and Stochastic Local Quantization. In Table 3, we show that JPEG compression as a defense does not significantly reduce model accuracies on low perturbation attacks like DF and CW-L2. For higher-perturbation attacks, the accuracy of Inception-v4 lowers by a maximum of 10%.

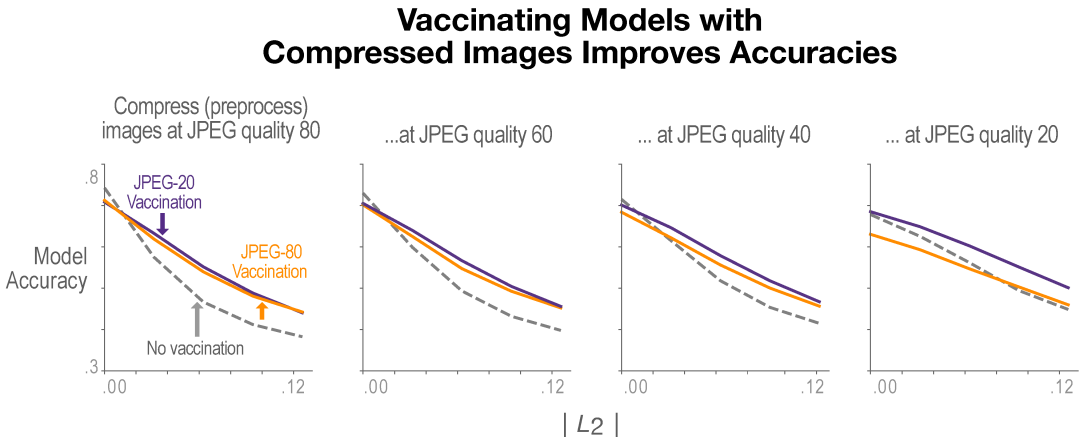

Figure 6: Vaccinating a model by retraining it with compressed images helps recover its accuracy. Each plot shows the model accuracies when preprocessing with different JPEG qualities with the FGSM attack. Each curve in the plot corresponds to a different model. The gray dotted curve corresponds to the original unvaccinated ResNet-v2 50 model. The orange and purple curves correspond to the models retrained on JPEG qualities 80 and 20 respectively. Retraining on JPEG compressed images and applying JPEG preprocessing helps recover accuracy in a gray-box attack.

4.5 NIPS 2017 Competition Results

In addition to the experiment results shown above, we also participated in the NIPS 2017 competition on Defense Against Adversarial Attack using a version of our approach that included JPEG compression and *vaccination* to defend against attacks "in the wild." With only an ensemble of three JPEG compression qualities (90, 80, 70), our entry received a silver badge in the competition, ranking 16th out of more than 100 submissions.

5 SIGNIFICANCE AND IMPACT

This work has been making multiple positive impacts on Intel's research and product development plans. In this section, we describe such impacts in detail, and also describe how they may more broadly influence deep learning and cybersecurity. We then discuss our work's scope, limitations, and additional practical considerations.

Ensemble	Cost	$\epsilon = 0$	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 6$	$\epsilon = 8$
$\mathcal{M}_q \times q$	64	73.90	67.72	60.13	54.44	49.84
$\mathcal{M}_q - q$	8	73.54	67.06	59.86	53.91	49.40
SHIELD	**4**	72.11	66.30	59.29	53.60	48.63

Table 2: Comparison of two ensemble schemes with SHIELD, when defending against FGSM. $\mathcal{M}_q \times q$ corresponds to each model $\mathcal{M}_q$ voting on each JPEG quality q from $q \in \{20, 30, 40, ..., 90\}$. In $\mathcal{M}_q - q$, each model $\mathcal{M}_q$ votes only on q, the JPEG quality it was trained on. SHIELD offers a favorable trade-off, providing at least 2x speed-up as compared to larger ensembles, while delivering comparable accuracies.

		Inc-v4 (80.2%)		**RN-v2 101** (77.0%)	
Attack	Defense	Accuracy	(Qual.)	Accuracy	(Qual.)
None	JPEG	79.05	(100)	76.48	(100)
	SLQ	75.90	-	73.70	-
CW-L2	JPEG	79.00	(100)	76.20	(100)
	SLQ	75.80	-	73.60	-
DF	JPEG	78.91	(100)	76.19	(100)
	SLQ	76.29	-	73.70	-
I-FGSM	JPEG	74.84	(100)	70.06	(70)
	SLQ	73.20	-	69.40	-
FGSM	JPEG	71.00	(100)	64.18	(40)
	SLQ	70.01	-	64.64	-

Table 3: JPEG compression as defense does not reduce model accuracy significantly on transferred attacks with low perturbation. Adversarial images crafted using the ResNet-v2 50 model are protected using *JPEG* alone and *Stochastic Local Quantization* (SLQ), before being fed into two other models: Inception-v4 (Inc-v4) and ResNet-v2 101 (RN-v2 101).

5.1 Software and Hardware Integration Milestones

As seen in Section 4, JPEG compression is much faster than other popular preprocessing techniques; even commodity implementations from Pillow are fast. However, in order to be deployed into a real defense pipeline, we need to evaluate its computational efficiency with tighter software and hardware integration. Fortunately, JPEG compression is a widely-used and mature technique that can be be easily deployed in various platforms, and due to its widespread usage, we can use off-the-shelf optimized software and hardware

for such testing. One promising milestone we reached, utilized Intel's hardware Quick Sync Video (QSV) technology: a hardware core dedicated and optimized for video encoding and decoding. It was introduced with Sandy Bridge CPU microarchitecture and exists currently in various Intel platforms. From our experiments, JPEG compression by Intel QSV is up to 24 times faster than the Pillow and TensorFlow implementations when evaluated on the same ImageNet validation set of 50,000 images. This computational efficiency is desirable for applications that need real-time defense, such as autonomous vehicles. In the future, we plan to explore the feasibility of our approach on more hardware platforms, such as the Intel Movidius Compute Stick[2], which is a low power USB-based deep learning inference kit.

5.2 New Computational Paradigm: Secure Deep Learning

This research has sparked insightful discussion with teams of Intel QSV, Intel Deep Learning SDK, and Intel Movidius Compute Stick. This work not only educates industry regarding concepts and defenses of adversarial machine learning, but also provides opportunities to advance deep learning software and hardware development to incorporate adversarial machine learning defenses. For example, almost all defenses incur certain levels of computational overhead. This may be due to image preprocessing techniques [14, 21], using multiple models for model ensembles [32], the introduction of adversarial perturbation detectors [22, 35], or the increase in training time for adversarial training [11]. However, while hardware and system improvement for fast deep learning training and inference remains an active area of research, secure machine learning workloads still receive relatively less attention, suggesting room for improvement. We believe this will accelerate the positive shift of thinking in the industry in the near future, from addressing problems like *"How do we build deep learning accelerators?"* to problems such as *"How do we build deep learning accelerators that are not only fast but also secure?"*. Understanding such hardware implications are important for microprocessor manufacturers, equipment vendors and companies offering cloud computing services.

5.3 Scope and Limitations

In this work, we focus on systematically studying the benefit of compression on its own. As myriads of newer and stronger attack strategies are continuously discovered, limitations in existing, single defenses are revealed. Our approach is not a panacea to defend all possible (future) attacks, and we do not expect or intend for it to be used in isolation of other techniques. Rather, our methods should be used together with other defense techniques, to potentially develop an even stronger defense. Using multi-layered protection is a proven, long-standing defense strategy that has been pervasive in security research and in practice [5, 34]. Fortunately, since our approach primarily involves preprocessing, it is easy to integrate it into many other defense techniques such as adversarial retraining.

[2]https://developer.movidius.com

6 RELATED WORK

Due to intriguing theoretical properties and practical importance, there has been a surge in the number of papers in the past few years attempting to find countermeasures against adversarial attacks. These include detecting adversarial examples before performing classification [10, 22], modifying network architecture and the underlying primitives used [13, 18, 29], modifying the training process [11, 26], and using preprocessing techniques to remove adversarial perturbations [3, 7, 14, 21]. The preprocessing approach is most relevant to our work. Below, we describe two methods in this category—median filter and total variation denoising, which we compared against in Section 4. We then discuss some recent attacks that claim to break preprocessing defenses.

6.1 Image Preprocessing as Defense

Median Filter. This method uses a sliding window over the image and replaces each pixel with the median value of its neighboring pixels to spatially smooth the image. The size of the the sliding window controls the smoothness, for example, a larger window size produces blurrier images. This technique has been used in multiple prior defense works [14, 35].

Total Variation Denoising. The method is based on the principle that images with higher levels of (adversarial) noise tend to have larger total variations: the sum of the absolute difference between adjacent pixel values. Denoising is performed by reducing the total variation while keeping the denoised image close to the original one. A weighting parameter is used as a trade-off between the level of total variation and the distance from the original image. Compared with median filter, this method is more effective at removing adversarial noise while preserving image details [14].

6.2 Attacks against Preprocessing Techniques

One of the reasons why adding preprocessing steps increases attack difficulty is that many preprocessing operations are non-differentiable, thus restricting the feasibility of gradient-based attacks. In JPEG compression, the quantization in the frequency domain is a non-differentiable operation.

Shin and Song [31] propose a method that approximates the quantization in JPEG with a differentiable function. They also optimize the perturbation over multiple compression qualities to ensure an adversarial image is robust at test time. However, the paper only reports preliminary results on 1000 images. It is also unclear whether their attack is effective against our more advanced SHIELD method, which introduces more randomization to combat against adversarial noise.

Backward Pass Differentiable Approximation [1] is another potential approach to bypass non-differentiable preprocessing techniques. To attack JPEG preprocessing, it performs forward propagation through the preprocessing and DNN combination but in the backward pass, the method differentiates with respect to the JPEG compressed image. This is based on the intuition that the compressed image should look similar to the original one, so the operation can be approximated by the identity function. However, we believe this assumption only holds for higher compression qualities. Since the work did not report the compression quality used in the experiments, the conclusion remains open for debate.

7 CONCLUSION

In this paper, we highlighted the urgent need for practical defense for deep learning models that can be readily deployed. We drew inspiration from JPEG image compression, a well-known and ubiquitous image processing technique, and placed it at the core of our new deep learning model defense framework: SHIELD. Since many attack strategies aim to perturb image pixels in ways that are visually imperceptible, the SHIELD defense framework utilizes JPEG compression to effectively "compress away" such pixel manipulation. SHIELD immunizes DNN models from being confused by compression artifacts by "vaccinating" a model: re-training it with compressed images, where different compression levels are applied to generate multiple vaccinated models that are ultimately used together in an ensemble defense. Furthermore, SHIELD adds an additional layer of protection by employing randomization at test time by compressing different regions of an image using random compression levels, making it harder for an adversary to estimate the transformation performed. This novel combination of vaccination, ensembling and randomization makes SHIELD a fortified multi-pronged defense, while remaining fast and successful without requiring knowledge about the model. We conducted extensive, large-scale experiments using the ImageNet dataset, and showed that our approaches eliminate up to 98% of gray-box attacks delivered by the recent, strongest attacks. To ensure reproducibility of our results, we have open-sourced our code on GitHub (https://github.com/poloclub/jpeg-defense).

ACKNOWLEDGEMENTS

This material is based in part upon work supported by the National Science Foundation under Grant Numbers IIS-1563816, CNS-1704701, and TWC-1526254. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This research is also supported in part by gifts from Intel, Google, Symantec, Yahoo! Labs, eBay, Amazon, and LogicBlox.

REFERENCES

[1] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *arXiv preprint arXiv:1802.00420* (2018).
[2] Anish Athalye and Ilya Sutskever. 2017. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397* (2017).
[3] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. 2017. Dimensionality Reduction as a Defense against Evasion Attacks on Machine Learning Classifiers. *arXiv preprint arXiv:1704.02654* (2017).
[4] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 39–57.
[5] Shang-Tse Chen, Yufei Han, Duen Horng Chau, Christopher Gates, Michael Hart, and Kevin A Roundy. 2017. Predicting Cyber Threats with Virtual Security Products. In *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 189–199.
[6] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. 2017. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900* (2017).
[7] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. 2016. A study of the effect of JPG compression on adversarial images. *arXiv preprint arXiv:1608.00853* (2016).
[8] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. 2017. Robust physical-world attacks on machine learning models. *arXiv preprint arXiv:1707.08945* (2017).
[9] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Dawn Song, Tadayoshi Kohno, Amir Rahmati, Atul Prakash, and Florian Tramer. 2017. Note on Attacking Object Detectors with Adversarial Stickers. *arXiv preprint arXiv:1712.08062* (2017).
[10] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. 2017. Detecting Adversarial Samples from Artifacts. *arXiv preprint arXiv:1703.00410* (2017).
[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. In *ICLR*.
[12] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2016. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435* (2016).
[13] Shixiang Gu and Luca Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068* (2014).
[14] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. 2018. Countering Adversarial Images using Input Transformations. *International Conference on Learning Representations* (2018).
[15] Weiwei Hu and Ying Tan. 2017. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. *arXiv preprint arXiv:1702.05983* (2017).
[16] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284* (2017).
[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
[18] Dmitry Krotov and John J Hopfield. 2017. Dense Associative Memory is Robust to Adversarial Inputs. *arXiv preprint arXiv:1701.00939* (2017).
[19] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).
[20] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. 2017. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. *arXiv preprint arXiv:1703.06748* (2017).
[21] Yan Luo, Xavier Boix, Gemma Roig, Tomaso Poggio, and Qi Zhao. 2015. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292* (2015).
[22] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On detecting adversarial perturbations. In *ICLR*.
[23] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In *CVPR*.
[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *CVPR*.
[25] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks Against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '17)*. 506–519.
[26] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*. 582–597.
[27] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The Limitations of Deep Learning in Adversarial Settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*. 372–387.
[28] Nicolas Papernot, Patrick D. McDaniel, Ananthram Swami, and Richard E. Harang. 2016. Crafting adversarial input sequences for recurrent neural networks. In *2016 IEEE Military Communications Conference, MILCOM*. 49–54.
[29] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. 2017. Improving Network Robustness against Adversarial Attacks with Compact Convolution. *arXiv preprint arXiv:1712.00699* (2017).
[30] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1528–1540.
[31] Richard Shin and Dawn Song. 2017. JPEG-resistant Adversarial Images. *NIPS 2017 Workshop on Machine Learning and Computer Security* (2017).
[32] Thilo Strauss, Markus Hanselmann, Andrej Junginger, and Holger Ulmer. 2017. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1709.03423* (2017).
[33] Christian Szegedy, Google Inc, Wojciech Zaremba, Ilya Sutskever, Google Inc, Joan Bruna, Dumitru Erhan, Google Inc, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.
[34] Acar Tamersoy, Kevin Roundy, and Duen Horng Chau. 2014. Guilt by association: large scale malware detection by mining file-relation graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1524–1533.
[35] Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Proceedings of the 2018 Network and Distributed Systems Security Symposium (NDSS)*.