ImageNet-Patch: A Dataset for Benchmarking Machine Learning Robustness against Adversarial Patches

Maura Pintor[a,c], Daniele Angioni[a], Angelo Sotgiu[a,c], Luca Demetrio[a,c], Ambra Demontis[a,*], Battista Biggio[a,c], Fabio Roli[b,c]

[a] *University of Cagliari, Italy*
[b] *University of Genova, Italy*
[c] *Pluribus One, Italy*

Abstract

Adversarial patches are optimized contiguous pixel blocks in an input image that cause a machine-learning model to misclassify it. However, their optimization is computationally demanding, and requires careful hyperparameter tuning, potentially leading to suboptimal robustness evaluations.

To overcome these issues, we propose ImageNet-Patch, a dataset to benchmark machine-learning models against adversarial patches. It consists of a set of patches, optimized to generalize across different models, and readily applicable to ImageNet data after preprocessing them with affine transformations. This process enables an approximate yet faster robustness evaluation, leveraging the transferability of adversarial perturbations. We showcase the usefulness of this dataset by testing the effectiveness of the computed patches against 127 models. We conclude by discussing how our dataset could be used

*Corresponding author
Email address: ambra.demontis@unica.it (Ambra Demontis)

as a benchmark for robustness, and how our methodology can be generalized to other domains. We open source our dataset and evaluation code at `https://github.com/pralab/ImageNet-Patch`.

Keywords: adversarial machine learning, adversarial patches, neural networks, defense, detection

1. Introduction

Understanding the security of machine-learning models is of paramount importance nowadays, as these algorithms are used in a large variety of settings, including security-related and mission-critical applications, to extract actionable knowledge from vast amounts of data. Nevertheless, such data-driven algorithms are not robust against attacks, as malicious attackers can easily alter the behavior of state-of-the-art models by carefully manipulating their input data [1, 2, 3, 4]. In particular, attackers can hinder the performance of classification algorithms by means of *adversarial patches* [5], i.e., contiguous chunks of pixels which can be applied to any input image to cause the target model to output an attacker-chosen class. When embedded into input images, adversarial patches produce out-of-distribution samples. The reason is that the injected patch induces a spurious correlation with the target label, which is likely to shift the input sample off the manifold of natural images. Adversarial patches can be printed as stickers and physically placed on real objects, like stop signs that are then recognized as speed limits [6], and accessories that camouflage the identity of a person, hiding their real

identity [7]. Therefore, the evaluation of the robustness against these attacks is of the uttermost importance, as they can critically impact real-world applications with physical consequences.

The only way to assess the robustness of a machine-learning system against adversarial patches is to generate and test them against the target model of choice. Adversarial patches are created by solving an optimization problem via gradient descent. However, this process is costly as it requires both querying the target model many times and computing the back-propagation algorithm until convergence is reached. Hence, it is not possible to obtain a fast robustness evaluation against adversarial patches without avoiding all the computational costs required by their optimization process. To further exacerbate the problem, adversarial patches should also be effective under different transformations, including affine transformations like translation, rotation and scale changes. This is required for patches to work also as attacks crafted in the physical world, where it is impossible to place them in a controlled manner, as well as to control the acquisition and environmental conditions. Moreover, adversarial patches should also be able to successfully transfer across different models, given that, in practical scenarios, it is most likely that complete access to the target model (i.e., access to its gradients), or the ability to query it for hundreds of times, is not provided.

To overcome these issues, in this work we propose ImageNet-Patch, a dataset of pre-optimized adversarial patches that can be used to benchmark

machine-learning models with small computational overhead. This dataset is constructed on top of a subset of the validation set of the ImageNet dataset. It consists of 10 patches that target 10 different classes, applied on $5,000$ images each, for a total of $50,000$ samples. We create these patches by solving an optimization process that includes an ensemble of models in its formulation, forcing the algorithm to propose patches that evade them all (Section 2). Considering an ensemble strengthens the effectiveness of our adversarial patches when used in transfer attacks since they gain generality and cross-model effectiveness. These patches are also manipulated with affine transformations during the optimization, to be invariant to rotation and translation, which makes them readily applicable in the physical world.

To use these patches as a benchmark, we then propose the following three-step approach, which is also depicted in Figure 1: (i) *initialization* of the samples, by extracting images from the ImageNet dataset; (ii) *dataset generation*, by applying the patches using random affine transformations; and (iii) *robustness evaluation* of the given models. Even though the resulting robustness evaluation will be approximate, this process is extremely simple and fast, and it provides a first quick step to evaluate the robustness of some newly-proposed defensive or robust learning mechanisms (Section 3).

We test the efficacy of ImageNet-Patch by evaluating 15 models that were not part of the initial ensemble as a test set, divided into 3 standard-trained models and 3 robustly-trained models, and we highlight the successful generalization of the patches to unseen models (Section 4). In addition, our results

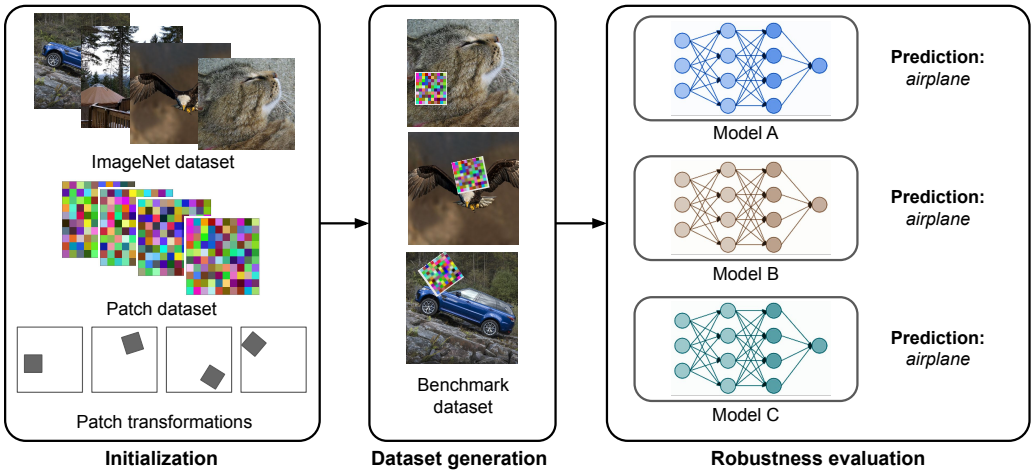

Figure 1: Robustness evaluation of target models with ImageNet-Patch: (i) *initialization*, we extract samples from the ImageNet dataset; (ii) *dataset generation*, we generate ImageNet-Patch by applying our adversarial patches to the selected images; and (iii) *robustness evaluation*, we compute the robustness of the target models by evaluating them against the ImageNet-Patch dataset.

demonstrate that this dataset can help evaluate the adversarial robustness and out-of-distribution performance of machine-learning models very quickly, without requiring one to solve cumbersome optimization problems. To foster reproducibility, we open-source the optimized patches along with the code used for evaluation.[1]

We conclude by discussing related work (Section 5), as well as the limitations and future directions of our work (Section 6), envisioning a leaderboard of machine-learning models based on their robustness to ImageNet-Patch.

[1] https://github.com/pralab/ImageNet-Patch

2. Crafting Transferable Adversarial Patches

Attackers can compute adversarial patches by solving an optimization problem with gradient-descent algorithms [5]. Since these patches are meant to be printed and attached to real-world objects, their effectiveness should not be undermined by the application of affine transformations, like rotation, translation and scale, that are unavoidable when dealing with this scenario. For example, an adversarial patch placed on a traffic sign should be invariant to scale changes to remain effective while an autonomous driving car approaches the traffic sign, or to camera rotation when taking pictures. Hence, the optimization process must include these perturbations as well, to force such invariance inside the resulting patches. Also, adversarial patches can either generate a general misclassification, namely an *untargeted* attack, or force the model to predict a specific class, namely a *targeted* attack. In this paper, we focus on the latter, and we consider a patch effective if it is able to correctly pilot the decision-making of a model toward an intended class.

More formally, targeted adversarial patches are computed by solving the following optimization problem:

$$\min_{\boldsymbol{\delta}} \mathbb{E}_{\mathbf{A}\sim\mathcal{T}} \left[\sum_{j=1}^{J} \mathcal{L}(\boldsymbol{x}_j \oplus \mathbf{A}\boldsymbol{\delta}, y_t; \boldsymbol{\theta}) \right], \tag{1}$$

where $\boldsymbol{\delta}$ is the adversarial patch to be computed, $\boldsymbol{x}_j$ is one of J samples of the

training data, y_t is the target label,[2] $\boldsymbol{\theta}$ is the targeted model, $\mathbf{A}$ is an affine transformation randomly sampled from a set of affine transformations $\mathcal{T}$, $\mathcal{L}$ is a loss function of choice, that quantifies the classification error between the target label and the predicted one and $\oplus$ is a function that applies the patch on the input images. The latter is defined as: $\boldsymbol{x} \oplus \boldsymbol{\delta} = (\mathbf{1} - \boldsymbol{\mu}) \odot \boldsymbol{x} + \boldsymbol{\mu} \odot \boldsymbol{\delta}$, where we introduce a mask $\boldsymbol{\mu}$ that is a tensor with the same size of the input data $\boldsymbol{x}$, and whose components are ones where the patch should be applied and zeros elsewhere [8]. This operator is still differentiable, as it is constructed by summing differentiable functions themselves; thus, it is straightforward to obtain the gradient of the loss function with respect to the patch.

To produce a dataset that can be used as a benchmark for an initial robustness assessment, with adversarial patches effective regardless of the target model, we can consider an ensemble of differentiable models inside the optimization process. This addition forces the optimization algorithm to find effective solutions against all the ensemble models, boosting the transferability of the produced adversarial patches. Namely, the ability of the adversarial patch optimized against a model (or a set of them) to be effective against different models. Hence, the loss function to be minimized can be written as:

$$\min_{\boldsymbol{\delta}} \mathbb{E}_{\mathbf{A} \sim \mathcal{T}} \left[\sum_{m=1}^{M} \sum_{j=1}^{J} \mathcal{L}(\boldsymbol{x}_j \oplus \mathbf{A}\boldsymbol{\delta}, y_t; \boldsymbol{\theta}_m) \right], \tag{2}$$

[2]The same formulation holds for crafting untargeted attacks, by simply substituting the target label y_t with the ground truth label of the samples y, and inverting the sign of the loss function.

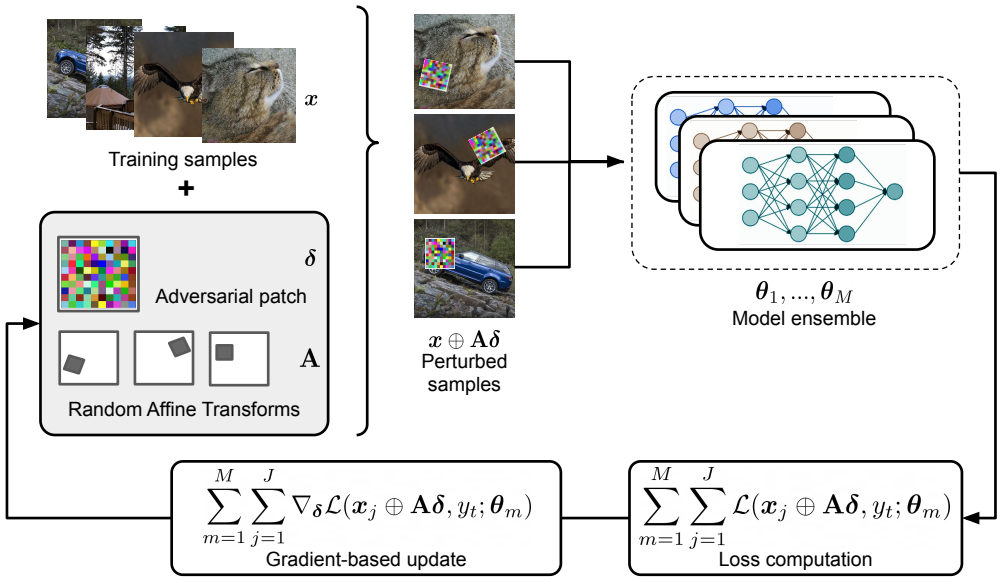

Figure 2: The optimization process, graphically described. At each step, we apply the patch to be optimized with random affine transformations on sample images, and we compute the scores of the ensemble. Hence, the algorithm computes the update step through gradient descent on the loss function w.r.t. the patch.

where we modified the original formulation in Equation 1 to minimize the loss $\mathcal{L}$ over a set of M models, respectively parameterized via $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M$.

The objective function defined in Equation 2 can be optimized through gradient-descent techniques, and thus we use Algorithm 1 for minimizing it. After having randomly initialized the patch (line 1), we loop through the number of intended epochs (line 2), and the samples of the training data (line 4). In each epoch, we sample a random affine transformation that will be applied to the patch (line 5). We iterate over all models of the ensemble (line 6) to calculate the loss by accumulating its gradient w.r.t. the patch (line 7), and using it to update the patch at the end of each epoch (line 8).

Algorithm 1 Optimization of adversarial patches on an ensemble of models

Input : $\boldsymbol{x}$, the training dataset containing J images; y_t, the target class; $\boldsymbol{\theta}_1, .., \boldsymbol{\theta}_M$, the ensemble of models; γ, the learning rate; N, the number of epochs.

Output: $\boldsymbol{\delta}$, the adversarial patch

1 $\boldsymbol{\delta} \sim U(0, 1)$ ▷ `Initialize patch with uniform distribution`
2 **for** $i \in [1, N]$ **do**
3 $\quad \boldsymbol{g} \leftarrow 0$ ▷ `Initialize gradient update for epoch` i
4 $\quad$ **for** $j \in [1, J]$ **do**
5 $\quad\quad \mathbf{A} \leftarrow$ `random-affine()` ▷ `Initialize transformation`
6 $\quad\quad$ **for** $m \in [1, M]$ **do**
7 $\quad\quad\quad \boldsymbol{g} \leftarrow \boldsymbol{g} + \frac{1}{MJ} \nabla_{\boldsymbol{\delta}} \mathcal{L}(\boldsymbol{x}_j \oplus \mathbf{A}\boldsymbol{\delta}, y_t; \boldsymbol{\theta}_m)$ ▷ `Accumulate gradients`
8 $\quad \boldsymbol{\delta} \leftarrow \boldsymbol{\delta} - \gamma \boldsymbol{g}$ ▷ `Optimize patch`
9 **return** $\boldsymbol{\delta}$ ▷ `Return optimized patch`

After all the epochs have been consumed, the final adversarial patch is returned (line 9). If the number of training samples is large, this algorithm can be easily generalized to a more efficient version using the gradient computed on a mini-batch to perform the updates, i.e. repeating the steps 3-8 for each batch of the training data. We present a graphical representation of our procedure in Figure 2.

3. The ImageNet-Patch Dataset

We now illustrate how we apply our methodology to generate the ImageNet-Patch dataset that will be used to evaluate the robustness of classification models against patch attacks.

The Baseline Dataset. We start from the validation set of the original Im-

ageNet database,[3] containing $1,281,167$ training images, $50,000$ validation images and $100,000$ test images, divided into $1,000$ object classes. From the validation set, we select a test set of $5,000$ images that matches exactly the ones used in RobustBench [9] for testing model robustness against adversarial attacks. This allows us not only to provide a direct comparison with the RobustBench framework, but also to easily add our benchmark to it. We create the corpus of images used to optimize adversarial patches from the remaining part of the ImageNet validation set, excluding the images used for the test set, and randomly sampling 20 images from different classes. Each patch is then optimized on these samples except the images of the target class of the attack. To clarify, if the attack is targeting the class "cup", we select one image for each of 20 different classes selected from the remaining 999 classes of the ImageNet dataset.

The ImageNet-Patch Dataset. We now define the ImageNet-Patch dataset. Since we optimize adversarial patches on an ensemble of chosen models, we select three deep neural network architectures trained on the ImageNet dataset, namely AlexNet [10], ResNet18 [11] and SqueezeNet [12]. We leverage the pretrained models available inside the PyTorch TorchVision zoo,[4] that are trained to take in input RGB images of size 224×224.

We run Algorithm 1 to create squared patches with a size of 50×50 pixels, with a learning rate of 1, 20 training samples selected as previously

[3] https://www.image-net.org/challenges/LSVRC/index.php
[4] https://pytorch.org/vision/master/models.html

Figure 3: The 10 optimized adversarial patches, along with their target labels.

described, 5000 training epochs, and using the cross-entropy as the loss function of choice. We consider rotation and translation as the applied affine transformations during the optimization of the patch, constraining rotations up to $\pm\frac{\pi}{8}$ to mimic the setup applied by Brown et al.[5], and translations to a shift of ± 68 pixels on both axes from the center of the image. The latter is a heuristic constraint, as we want to avoid corner cases where the adversarial patch is too close to the boundaries of the image. We also keep the size of the adversarial patch fix to 50×50 pixels during the optimization process.

We optimize 10 different patches with these settings, targeting 10 different classes of the ImageNet dataset ("soap dispenser", "cornet", "plate", "banana", "cup", "typewriter keyboard", "electric guitar", "hair spray", "sock", "cellular phone"). The resulting patches are shown in Figure 3. We apply such patches to each of the $5,000$ images in the test set along with random affine transformations, generating a dataset of $50,000$ perturbed images

with adversarial patches. We depict some examples of the applied patches in Figure 4.

4. Experimental Analysis

We now showcase experimental results related to the robustness evaluation through the usage of our ImageNet-Patch dataset. We first explain the metrics (Section 4.1), and which models we consider for evaluating our dataset (Section 4.2). We then proceed in detailing the results of our experiments (Section 4.3), by considering the previously introduced metrics, and lastly we show the same measurements but extended to a large-scale model selection (Section 4.4).

4.1. Evaluation Metrics

We evaluate the evasion performance of the ImageNet-Patch dataset by considering three metrics: (i) the *clean accuracy*, which is the accuracy of the target model in absence of attacks; (ii) the *robust accuracy*, which is the accuracy of the target model in presence of adversarial patches; and (iii) the *success rate* of a patch, that measures the percentage of samples for which the patch successfully altered the prediction of the target model toward the intended class.

Clean Accuracy. We denote with the operator $\mathcal{A}_k(\boldsymbol{x}, y; \boldsymbol{\theta})$ the top-k accuracy, i.e. by inspecting if the desired class y appears in the set of k highest outputs of the classification model $\boldsymbol{\theta}$ when receiving the sample x as input. We then use this operator for defining the clean accuracy C_k, as:

Figure 4: A batch of clean images initially predicted correctly by a SqueezeNet [12] model, and its perturbation with 5 different adversarial patches. Each row contains the original image with a different patch, whose target is displayed in the left. The predictions are shown on top of each of the samples, in *green* for correct prediction, *blue* for misclassification, and in *red* for a prediction that ends up in the target class of the attack.

$$C_k = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}_{test}} \left[\mathcal{A}_k(\boldsymbol{x}, y; \boldsymbol{\theta})\right], \tag{3}$$

and the other metrics that we use for our experimental evaluation.

Robust Accuracy. We define the value R_k as the top-k accuracy on the images after the application of the patch with the random rototranslation transformations, formalized as:

$$R_k = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},y)\sim\mathcal{D}_{test} \\ \mathbf{A}\sim\mathcal{T}}} \left[\mathcal{A}_k(\boldsymbol{x} \oplus \mathbf{A}\boldsymbol{\delta}, y; \boldsymbol{\theta})\right] \tag{4}$$

Success Rate. We define the value S_k as the success rate of the attack, i.e. the top-k accuracy on the target label y_t instead of the ground truth label y, formalized as:

$$S_k = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{x},y)\sim\mathcal{D}_{test} \\ \mathbf{A}\sim\mathcal{T}}} \left[\mathcal{A}_k(\boldsymbol{x} \oplus \mathbf{A}\boldsymbol{\delta}, y_t; \boldsymbol{\theta})\right] \tag{5}$$

We evaluate these three metrics for $k = 1, 5, 10$.

4.2. Evaluation Protocol

To evaluate the effectiveness of the patches, we test our ImageNet-Patch dataset against 127 deep neural networks trained on the ImageNet dataset. To facilitate the discussion, we group the models in 5 groups, namely the `ENSEMBLE`, `STANDARD`, `ADV-ROBUST`, `AUGMENTATION`, `MORE-DATA` groups. In a first analysis, we consider 15 models to discuss results in detail, and further extend the analysis with a large-scale analysis, presented in Section 4.4. In

particular, we consider the three models used for the ensemble, AlexNet [10], ResNet18 [11] and SqueezeNet [12], as the first group, `ENSEMBLE`. We consider for the second group, `STANDARD`, 3 standard-trained models, that are GoogLeNet [13], MobileNet [14] and Inception v3 [15], available in PyTorch Torchvision. We then consider 3 robust-trained models as the `ADV-ROBUST` available on RobustBench, specifically a ResNet-50 proposed by Salman et al. [16], a ResNet-50 proposed by Engstrom et al. [17] and a ResNet-50 proposed by Wong et al. [18]. We also additionally consider a set of 6 models from the ImageNet Testbed repository[5] proposed by Taori et al. [19], to analyze the effects of non-adversarial augmentation techniques and of training on bigger datasets. We select 3 models specifically trained for being robust to common image perturbations and corruptions, namely the models proposed by Zhang et al. [20], Hendrycks et al [21], and Engstrom et al [22], that we group as `AUGMENTATION` group. We further select other 3 models, namely two of the ones proposed by Yalniz et al. [23] and one proposed by Mahajan et al. [24], that have been trained on datasets that utilize substantially more training data than the standard ImageNet training set. We group these last models as the `MORE-DATA` group. Lastly, the `STANDARD`, `ADV-ROBUST`, `AUGMENTATION`, and `MORE-DATA` groups will be referred as the *Unknown* models, since they are not used while optimizing the adversarial patches.

[5] https://github.com/modestyachts/imagenet-testbed

4.3. Experimental Results

We now detail the effectiveness of our dataset against the groups we have isolated, by evaluating them with the chosen metrics, sharing the results in Table 1 and Figure 5, where we confront the relation between clean and robust accuracy, and also the relation between robust accuracy and success rate.

Evaluation of Known Models. The `ENSEMBLE` group of models is characterized by low robust accuracy and the highest success rate of the adversarial patch. Such a result is trivially intuitive since we optimize our adversarial patches to specifically mislead these models, as they are part of the training ensemble.

Evaluation of Unknown Models. These models are not part of the ensemble used to optimize the adversarial patches. First of all, all of them highlight a good clean accuracy on our clean test set of images.

The `STANDARD` group is characterized by a modest decrement of the robust accuracy, highlighting errors caused by the patches. The success rate is lower compared to those exhibited by the `ENSEMBLE` group, since patches are not optimized on these models, but it raises considerably when considering different top-k results. Hence, even if the adversarial patches are sometimes unable to target precisely one class, they are still rising its confidence among the scores.

The `ADV-ROBUST` group is characterized by a drop of robust accuracy similar to the `STANDARD` group, but with an almost-zero success rate for the

adversarial patches. This implies that robust models are affected by adversarial patches in terms of untargeted misclassifications, but not by targeted ones.

The AUGMENTATION group contains mixed results, shifting from a modest to a severe drop in terms of robust accuracy, associated with an increment of the success rate, which is slightly less than that achieved by the STANDARD group. This might imply that augmentation techniques help the model to score good results on regular images, but performance drops when dealing with adversarial noise.

Lastly, the MORE-DATA group scores the best in terms of both clean and robust accuracy while the success rate of the adversarial patches is similar to the AUGMENTATION group results.

4.4. Large-scale Analysis

We now discuss the effectiveness of our dataset on a large-scale setting, where we extend the analysis to a pool of 127 models, including also the ones already tested in Section 4.3. These are all the models available in RobustBench [9] and in ImageNet Testbed [19], again divided into the same groups (STANDARD, ADV-ROBUST, AUGMENTATION and MORE-DATA). We plot our benchmark in Figure 6, confirming the results presented in Section 4.3. To better highlight the efficacy of our adversarial patches, we also depict the difference in terms of accuracy of these target models scored by applying our pre-optimized patches and randomly-generated ones in Figure 7. The top

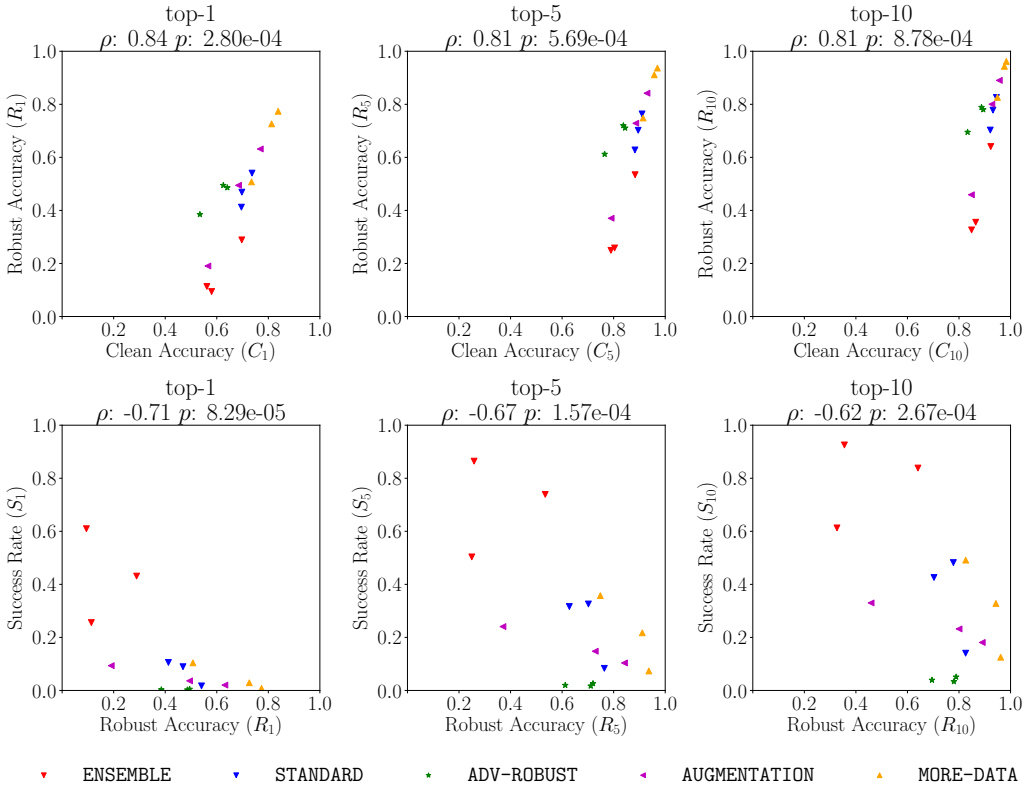

Figure 5: Analysis for results shown in Table 2. *Top row*: top-1 (left), top-5 (center), and top-10 (right) clean accuracy vs robust accuracy. *Bottom row*: top-1 (left), top-5 (center), and top-10 (right) robust accuracy vs attack success rate. The Pearson correlation coefficient ρ and the p-value are also reported for each plot.

row shows the results for the pre-optimized patches, while the bottom row focuses on the random ones, and each plot also shows a robust regression line, along with its 95% confidence interval.

The regression we compute on our metrics highlights meaningful observations we can extract from the benchmark. First, the robust accuracy of each model evaluated with random patches can be still computed as a linear function of clean accuracy, as shown by the plot of the second row of Figure 7.

		top-1			top-5			top-10		
	Model	C_1	R_1	S_1	C_5	R_5	S_5	C_{10}	R_{10}	S_{10}
ENSEMBLE	AlexNet [10]	0.562	0.113	0.256	0.789	0.250	0.504	0.849	0.327	0.613
	ResNet18 [11]	0.697	0.289	0.431	0.883	0.535	0.739	0.923	0.641	0.839
	SqueezeNet [12]	0.580	0.094	0.610	0.804	0.259	0.865	0.865	0.355	0.926
STANDARD	GoogLeNet [13]	0.697	0.469	0.090	0.895	0.702	0.326	0.932	0.778	0.482
	MobileNet [14]	0.737	0.541	0.017	0.910	0.764	0.083	0.945	0.826	0.141
	Inception v3 [15]	0.696	0.412	0.106	0.883	0.628	0.317	0.921	0.703	0.426
ADV-ROBUST	Engstrom et al. [17]	0.625	0.495	0.005	0.838	0.720	0.026	0.887	0.789	0.051
	Salman et al. [16]	0.641	0.486	0.003	0.845	0.711	0.017	0.894	0.780	0.034
	Wong et al. [18]	0.535	0.385	0.003	0.765	0.612	0.020	0.833	0.695	0.039
AUGM.	Zhang et al. [20]	0.566	0.191	0.093	0.790	0.370	0.241	0.848	0.459	0.330
	Hendrycks et al [21]	0.769	0.632	0.020	0.929	0.842	0.104	0.956	0.890	0.181
	Engstrom et al [22]	0.684	0.495	0.036	0.886	0.729	0.148	0.928	0.800	0.232
MORE-DATA	Yalniz et al. [23]-a	0.813	0.726	0.029	0.958	0.911	0.217	0.976	0.943	0.328
	Yalniz et al. [23]-b	0.838	0.774	0.008	0.970	0.936	0.073	0.984	0.962	0.125
	Mahajan et al. [24]	0.735	0.507	0.104	0.914	0.748	0.357	0.949	0.826	0.491

Table 1: Evaluation of the ImageNet-Patch dataset using the chosen metrics, as described in Section 4.2. On the rows, we list the 15 models used for testing, divided into the isolated groups. On the columns, we detail the clean accuracy, the robust accuracy and the success rate of the adversarial patch, repeated for top-1,5, and 10 accuracy.

Hence, the clean accuracy can be seen as an accurate estimator of the robust accuracy when using random patches, similarly to what has been found by Taori et al. [19]. However, when we evaluate the robustness with our pre-optimized patches, the relation between robust and clean accuracy slightly diverges from a linear regression model, as the distance of the points from the interpolating line increases. Such effect is also enforced by the Pearson correlation computed and reported on top of each plot, since it is lower when

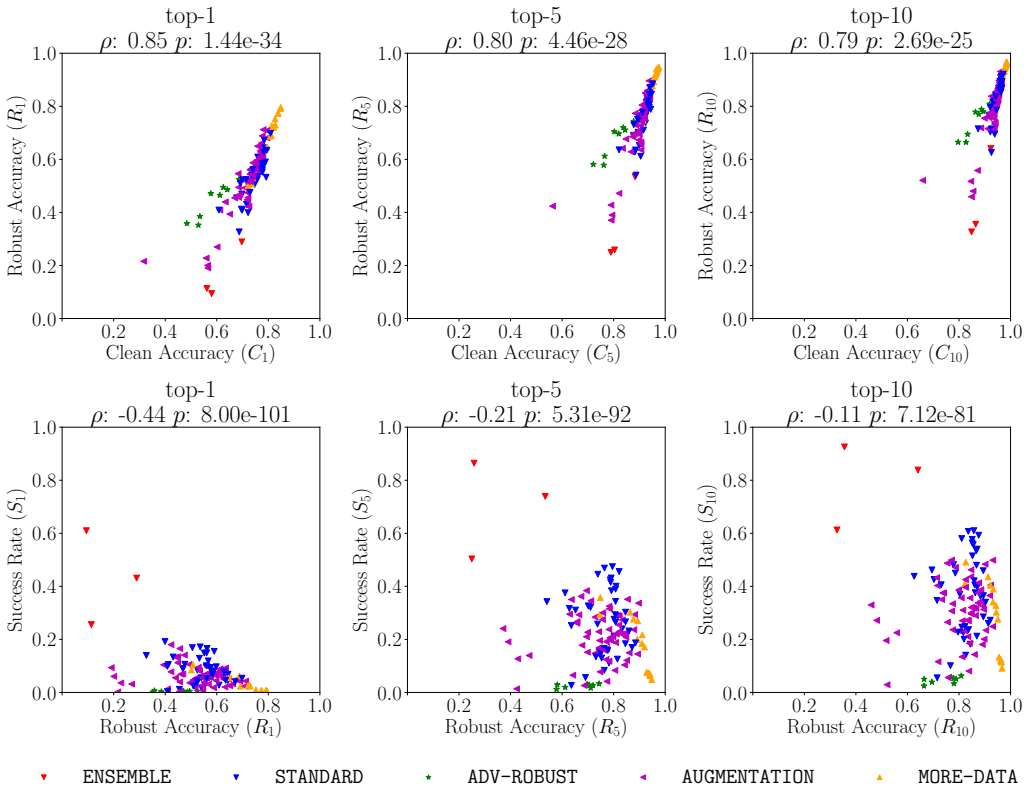

Figure 6: Results of our large-scale analysis on 127 publicly-released models. *Top row*: top-1 (left), top-5 (center), and top-10 (right) clean accuracy vs robust accuracy. *Bottom row*: top-1 (left), top-5 (center), and top-10 (right) robust accuracy vs attack success rate. The Pearson correlation coefficient ρ and the p-value are also reported for each plot.

using adversarial patches.

Among the many reasons behind this effect, we focus on the `ADV-ROBUST` group, as it lays outside the confidence level, and towards the bisector of the plot, lowering for sure the computed correlation. Intuitively, models that are located above the regression line can be considered more robust when compared with the others, since their robust accuracy is closer to their clean accuracy, i.e. closer to the bisector line. However, even if their robust training

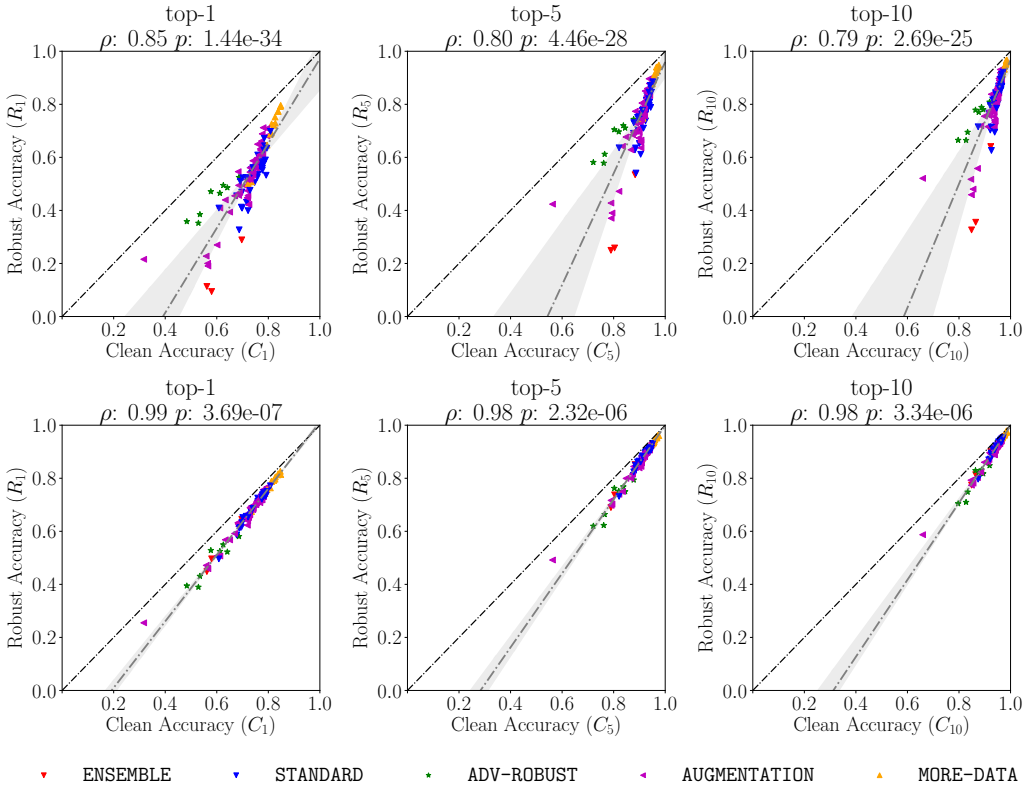

Figure 7: Clean vs robust accuracy for adversarial (*top row*) and random (*bottom row*) patches. The Pearson correlation coefficient ρ and the p-value are also reported for each plot. The dashed grey line and shaded area show a robust regression model fitted on the data along with the 95% confidence intervals. The results highlight the effectiveness of our pre-optimized strategy against choosing patches at random.

is aiding their performances against patch attacks, their robustness is not as evident as the one obtained when considering their original threat model. Evaluating adversarial robustness on limited threat models is therefore not sufficient to have a clear idea of what impact attacks can have on these models. Our dataset can help by providing additional analysis of robustness against patch attacks to assess for a more general and complete evaluation.

Lastly, we notice that the `MORE-DATA` group seems to present a similar

effect by distantiating from the regression line, but with a much lower magnitude. The effect is less evident because these models start from a higher clean accuracy, which then leads to a naturally higher robust accuracy.

4.5. Discussion

We briefly summarize here the results of our analysis, based on our ImageNet-Patch dataset to benchmark machine-learning models. We observe that data augmentation techniques do not generally improve robustness to adversarial patches. Moreover, we argue that real progress in robustness should be observed as a general property against different adversarial attacks, and not only against one specific perturbation model with a given budget (e.g., ℓ_∞-norm perturbations with maximum size of 8/255). Considering defenses that work against one specific perturbation model may be too myopic and hinder sufficient progress in this area. We are not claiming that work done on defenses for adversarial attacks so far is useless. Conversely, there has been great work and progress in this area, but it seems now that defenses are becoming too specific to current benchmarks and fail to generalize against slightly-different perturbation models. To overcome this issue, we suggest to test the proposed defenses on a wider set of robustness benchmarks, rather than over-specializing them on a specific scenario, and we do believe that our ImageNet-Patch benchmark dataset provides a useful contribution in this direction.

Attack	Cross-model	Transfer	Targeted	Untargeted	Transformations
Sharif et al. [7]	✗	✗	✓	✓	`rot`
Brown et al. [5]	✓	✓	✓	✗	`loc`, `scl`, `rot`
LaVAN [8]	✗	✗	✓	✗	`loc`
PS-GAN [25]	✗	✓	✗	✓	`loc`
DT-Patch [26]	✗	✗	✓	✗	✗
PatchAttack [27]	-	✓	✓	✓	`loc`, `scl`
IAPA [28]	✗	✓	✓	✓	✗
Lennon et al. [29]	✗	✓	✓	✗	`loc`, `scl`, `rot`
Xiao et al. [30]	-	✓	✓	✓	`various`
Ye et al. [31]	✓	✓	✓	✗	`loc`, `scl`, `rot`
GDPA [32]	✗	✗	✓	✓	`loc`
Ours	✓	✓	✓	✓	`loc`, `rot`

Table 2: Patch attacks, compared based on their main features. `loc` refers to the location of the patch in the image, `rot` refers to rotation, `scl` refers to scale variations, `various` include several image transformations (see [30] for more details).

5. Related Work

We now discuss relevant work related to the optimization of adversarial patches, and to the proposal of similar benchmark datasets.

5.1. Patch Attacks

The first physical attack against deep neural networks was proposed by [7], by developing an algorithm for printing adversarial eyeglass frames able to evade a face recognition system. Brown et al. [5] introduced the first universal patch attack that focuses on creating a physical perturbation. Such is obtained by optimizing patches on an ensemble of models to achieve targeted misclassification when applied to different input images with different transformations. The LaVAN attack, proposed by [8], attempts to achieve the same goal of Brown et al. by also reducing the patch size by placing it in regions of the target image where there are no other objects.

The PS-GAN attack, proposed by Liu et al. [25], addresses the problem of minimizing the perceptual sensitivity of the patches by enforcing visual fidelity while achieving the same misclassification objective. The DT-Patch attack, proposed by Benz et al. [26], focuses on finding universal patches that only redirect the output of some given classes to different target labels, while retaining normal functioning of the model on the other classes. PatchAttack, proposed by Yang et al. [27], leverages reinforcement learning for selecting the optimal patch position and texture to use for perturbing the input image for targeted or untargeted misclassification, in a black-box setting. The Inconspicuous Adversarial Patch Attack (IAPA), proposed by Bai et al. [28], generates difficult-to-detect adversarial patches with one single image by using generators and discriminators. Lennon et al. [29] analyze the robustness of adversarial patches and their invariance to 3D poses. Xiao et al. [30] craft transferable patches using a generative model to fool black-box face recognition systems. They use the same transformations as [33], but unlike other attacks, they apply them to the input image with the patch attached, and not just on the patch. Ye et al. [31] study the specific application of patch attacks on traffic sign recognition and use an ensemble of models to improve the attack success rate. The Generative Dynamic Patch Attack (GDPA), proposed by Li et al. [32], generates the patch pattern and location for each input image simultaneously, reducing the runtime of the attack and making it hence a good candidate to use for adversarial training.

We summarize in Table 2 these attacks, highlighting the main proper-

ties and comparing them with the attack we used to create the adversarial patches. In particular, in the *Cross-model* column we report the capability of an attack to be performed against multiple models (for black-box attacks we omit this information); in the *Transfer* column the proved transferability of patches, if reported in each work (thus it is still possible that an attack could produce transferable patches even if not tested on this setting); in *Targeted* and *Untargeted* columns the type of misclassification that patches can produce; in *Transformations* column the transformations applied to the patch during the optimization process (if any), which can increase the robustness of the patches with respect to them at test time.

In this work, we leverage the model-ensemble attack proposed by Brown et al. [5] to create adversarial patches that are robust to affine transformations and that can be applied to different source images to cause misclassification on different target models. From that, we publish a dataset that favors fast robustness evaluation to patch attacks without requiring costly steps for the optimization of the patches.

5.2. Benchmark Datasets for Robustness Evaluations

Previous work proposed datasets for benchmarking adversarial robustness. The APRICOT dataset, proposed by Braunegg et al. [34], contains $1,000$ annotated photographs of printed adversarial patches targeting object detection systems, i.e. producing targeted detections. The images are collected in public locations and present different variations in position, dis-

tance, lighting conditions, and viewing angle. ImageNet-C and ImageNet-P, proposed by Hendrycks et al. [35], are two datasets proposed to benchmark neural network robustness to image corruptions and perturbations, respectively. ImageNet-C perturbs images from the ImageNet dataset with a set of 75 algorithmically-generated visual corruptions, including noise, blur, weather, and digital categories, with different strengths. ImageNet-P perturbs images again from the ImageNet dataset and contains a sequence of subtle perturbations that slowly perturb the image to assess the stability of the networks' prediction on increasing amounts of perturbations.

Differently from these works, we propose a dataset that can be used to benchmark the robustness of image classifiers to adversarial patch attacks, whose aim is not restricted to being a source used at training time to improve robustness, or a collection of environmental corruptions.

6. Conclusions, Limitations, and Future Work

We propose the ImageNet-Patch dataset, a collection of pre-optimized adversarial patches that can be used to compute an approximate-yet-fast robustness evaluation of machine-learning models against patch attacks. This dataset is constructed by optimizing squared blocks of contiguous pixels perturbed with affine transformations to mislead an ensemble of differentiable models, forcing the optimization algorithm to produce patches that can transfer across models, gaining cross-model effectiveness. Finally, these adversarial patches are attached to images sampled from the ImageNet dataset, compos-

ing a benchmark dataset of 50,000 images. The latter is used to make an initial robustness evaluation of a selected pool of both standard-trained and robust-trained models, disjointed from the ensemble used to optimize the patch, showing that our methodology is already able to decrease their performances with very few computations needed. The latter highlights the need of considering a wider scope when evaluating adversarial robustness, since the latter should be a general property and not customized on single strategies. Hence, our dataset can be used to bridge this gap, and to rapidly benchmark the adversarial robustness and out-of-distribution performance of machine-learning models for image classification.

Limitations. While our methodology is efficient, it only provides an approximated evaluation of adversarial robustness, which can be computed more accurately by performing adversarial attacks against the target model, instead of using transfer attacks. Hence, our analysis serves as a first preliminary robustness evaluation, to highlight the most promising defensive strategies. Moreover, we only release patches that target 10 different classes, and this number could be extended to target all the 1000 classes of the ImageNet dataset.

Future work. We envision the use of our ImageNet-Patch dataset as a benchmark for machine-learning models, which may be added to the RobustBench project, where recently-proposed robust models are evaluated against an attack test suite and then ranked w.r.t. their robustness. In addition, by tuning the algorithms, our methodology can, in theory, generate adversarial

patches for any kind of datasets of images, extending the achieved results on ImageNet to other data sources as well. We finally argue that these pre-optimized adversarial patches might provide some benefit when used as an initialization point for other attacking strategies and later fine-tuned to save time and computations.

7. Acknowledgements

This work was partly supported by the PRIN 2017 project RexLearn, funded by the Italian Ministry of Education, University and Research (grant no. 2017TWNMH2); by BMK, BMDW, and the Province of Upper Austria in the frame of the COMET Programme managed by FFG in the COMET Module S3AI.

References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: H. Blockeel, K. Kersting, S. Nijssen, F. Železný (Eds.), Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part III, Vol. 8190 of LNCS, Springer Berlin Heidelberg, 2013, pp. 387–402.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations, 2014.

[3] N. Carlini, D. A. Wagner, Towards evaluating the robustness of neural networks, in: IEEE Symposium on Security and Privacy, IEEE Computer Society, 2017, pp. 39–57.

[4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations, 2018.

[5] T. B. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer, Adversarial patch, arXiv preprint arXiv:1712.09665 (2017).

[6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust physical-world attacks on deep learning visual classification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1625–1634.

[7] M. Sharif, S. Bhagavatula, L. Bauer, M. K. Reiter, 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, 2016, pp. 1528–1540.

[8] D. Karmon, D. Zoran, Y. Goldberg, Lavan: Localized and visible adversarial noise, in: International Conference on Machine Learning, PMLR, 2018, pp. 2507–2515.

[9] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammar-

ion, M. Chiang, P. Mittal, M. Hein, Robustbench: a standardized adversarial robustness benchmark, arXiv preprint arXiv:2010.09670 (2020).

[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).

[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[12] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡ 0.5 mb model size, arXiv preprint arXiv:1602.07360 (2016).

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 1–9.

[14] A. G. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, H. Adam, Searching for mobilenetv3, 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 1314–1324.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the

inception architecture for computer vision, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2818–2826.

[16] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, A. Madry, Do adversarially robust imagenet models transfer better?, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.
URL https://proceedings.neurips.cc/paper/2020/hash/24357dd085d2c4b1a88a7e0692e60294-Abstract.html

[17] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, D. Tsipras, Robustness (python library) (2019).
URL https://github.com/MadryLab/robustness

[18] E. Wong, L. Rice, J. Z. Kolter, Fast is better than free: Revisiting adversarial training, in: International Conference on Learning Representations, 2020.
URL https://openreview.net/forum?id=BJx040EFvH

[19] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, L. Schmidt, Measuring robustness to natural distribution shifts in image classification, Advances in Neural Information Processing Systems 33 (2020) 18583–18599.

[20] R. Zhang, Making convolutional networks shift-invariant again, in: ICML, 2019.

[21] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, J. Gilmer, The many faces of robustness: A critical analysis of out-of-distribution generalization, ICCV (2021).

[22] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, A. Madry, Exploring the landscape of spatial robustness, in: International Conference on Machine Learning, 2019, pp. 1802–1811.

[23] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, D. Mahajan, Billion-scale semi-supervised learning for image classification (2019). arXiv:1905.00546.

[24] D. K. Mahajan, R. B. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, L. van der Maaten, Exploring the limits of weakly supervised pretraining, in: ECCV, 2018.

[25] A. Liu, X. Liu, J. Fan, Y. Ma, A. Zhang, H. Xie, D. Tao, Perceptual-sensitive gan for generating adversarial patches, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 33, 2019, pp. 1028–1035.

[26] P. Benz, C. Zhang, T. Imtiaz, I. S. Kweon, Double targeted universal adversarial perturbations, in: Proceedings of the Asian Conference on Computer Vision, 2020.

[27] C. Yang, A. Kortylewski, C. Xie, Y. Cao, A. Yuille, Patchattack: A black-box texture-based attack with reinforcement learning, in: European Conference on Computer Vision, Springer, 2020, pp. 681–698.

[28] T. Bai, J. Luo, J. Zhao, Inconspicuous adversarial patches for fooling image recognition systems on mobile devices, IEEE Internet of Things Journal (2021).

[29] M. Lennon, N. Drenkow, P. Burlina, Patch attack invariance: How sensitive are patch attacks to 3d pose?, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 112–121.

[30] Z. Xiao, X. Gao, C. Fu, Y. Dong, W. zhe Gao, X. Zhang, J. Zhou, J. Zhu, Improving transferability of adversarial patches on face recognition with generative models, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 11840–11849.

[31] B. Ye, H. Yin, J. Yan, W. Ge, Patch-based attack on traffic sign recognition, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2021, pp. 164–171.

[32] X. Li, S. Ji, Generative dynamic patch attack, arXiv preprint arXiv:2111.04266 (2021).

[33] C. Xie, Z. Zhang, J. Wang, Y. Zhou, Z. Ren, A. L. Yuille, Improving transferability of adversarial examples with input diversity, 2019

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2725–2734.

[34] A. Braunegg, A. Chakraborty, M. Krumdick, N. Lape, S. Leary, K. Manville, E. Merkhofer, L. Strickhart, M. Walmer, Apricot: A dataset of physical adversarial attacks on object detection, in: European Conference on Computer Vision, Springer, 2020, pp. 35–50.

[35] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, in: International Conference on Learning Representations, 2018.