Jedi: Entropy-based Localization and Removal of Adversarial Patches

Bilel Tarchoun [*], Anouar Ben Khalifa [*,⊥], Mohamed Ali Mahjoub [*], Nael Abu-Ghazaleh [†]
Ihsen Alouani [‡,ϒ]
[‡] CSIT, Queen's University Belfast, UK
[*] Université de Sousse, Ecole Nationale d'Ingénieurs de Sousse, LATIS, Sousse, Tunisia;
[ϒ] IEMN CNRS 8520, Université Polytechnique Hauts-de-France
[⊥] Université de Jendouba, Institut National des Technologies et des Sciences du Kef, Tunisia;
[†] University of California Riverside, CA, USA
bilel.tarchoun@eniso.u-sousse.tn,i.alouani@qub.ac.uk

Abstract

Real-world adversarial physical patches were shown to be successful in compromising state-of-the-art models in a variety of computer vision applications. Existing defenses that are based on either input gradient or features analysis have been compromised by recent GAN-based attacks that generate naturalistic patches. In this paper, we propose Jedi*, a new defense against adversarial patches that is resilient to realistic patch attacks. Jedi tackles the patch localization problem from an information theory perspective; leverages two new ideas: (1) it improves the identification of potential patch regions using* ***entropy analysis****: we show that the entropy of adversarial patches is high, even in naturalistic patches; and (2) it improves the localization of adversarial patches, using an autoencoder that is able to complete patch regions from high entropy kernels. Jedi achieves high-precision adversarial patch localization, which we show is critical to successfully repair the images. Since Jedi relies on an input entropy analysis, it is* model-agnostic*, and can be applied on pre-trained off-the-shelf models without changes to the training or inference of the protected models. Jedi detects on average* 90% *of adversarial patches across different benchmarks and recovers up to* 94% *of successful patch attacks (Compared to 75% and 65% for LGS and Jujutsu, respectively).*

1. Introduction

Deep neural networks (DNNs) are vulnerable to adversarial attacks [20] where an adversary adds carefully crafted imperceptible perturbations to an input (e.g., by l_p-norm bounded noise magnitude), forcing the models to misclassify. Several adversarial noise generation methods have been proposed [4, 10], often as part of a cat and mouse game where new defenses emerge [1, 18] only to be shown vulnerable to new adaptive attacks. Under real-world conditions, an attacker creates a physical patch (thus, spatially constrained) that contains an adversarial pattern. Such a patch can be placed as a sticker on traffic signs [9], worn as part of an item of clothing [13, 21], or introduced using a display monitor [12], providing a practical approach for attackers to carry out adversarial attacks. First proposed by Brown et al. [3], these patches are different from traditional adversarial attacks in two primary ways: (1) they occupy a constrained space within an image; and (2) they may not be noise budget-constrained within the patch. Several adversarial patch generation methods have been demonstrated [13, 14, 16, 21], many of which showcasing real-life implementation, making them an ongoing threat to visual ML systems.

Several approaches aiming to detect adversarial patches and defuse their impact have been proposed [5, 11, 15, 17, 22, 24, 25]. One category of defenses attempts to locate patches by detecting anomalies caused by the presence of the patch. These anomalies can be identified in the input pixel data such as in the case of Localized Gradient Smoothing [17] where the patch is located using high pixel gradient values. Alternatively, they can be identified in the feature space where the adversarial patch can create irregular saliency maps with regards to its targeted class that can be exploited by defenses such as in Digital Watermarking [11] and Jujutsu [5]. These defenses have two primary limitations: (1) They are only moderately successful against baseline attacks enabling recovery from many attacks (e.g., 75% for LGS and 65% for Jujutsu); and (2) they are vulnerable to adaptive attacks that generate *naturalistic* adversarial patches that are meant to use patterns similar to natural images. Hu et al. [13] train a GAN to generate naturalistic

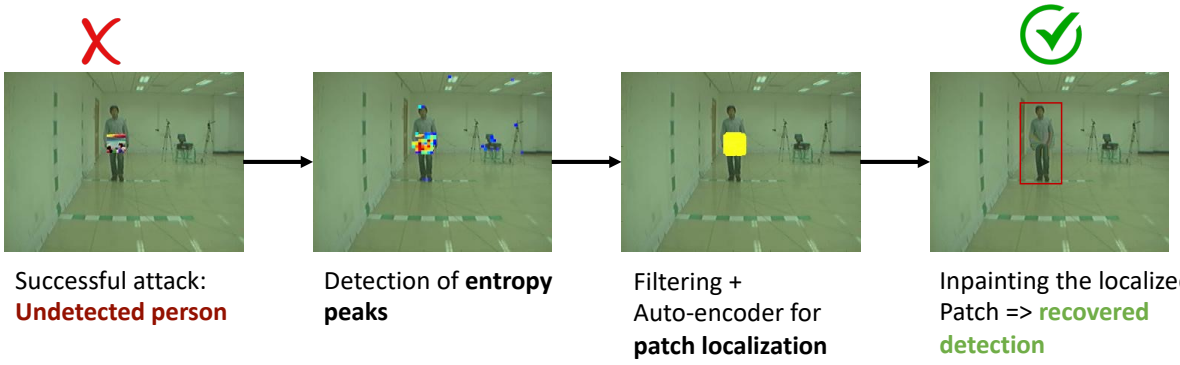


Figure 1. Illustration on a patch attack against Yolo. Jedi surgically localizes the patch via entropy analysis and recovers the image.

patches, matching the visual properties of normal images, and show that they are able to bypass defenses that are based on either input or features analysis.

In this paper, we propose *Jedi*[1], a new defense approach that surgically localizes adversarial patches to finally reconstruct the initial image. Compared to state of the art, *Jedi* enables both more accurate detection and recovery from patches, as well as resilience to adaptive attacks leveraging two primary ideas: (1) Using **entropy** to improve the identification of suspicious regions of the image: we show that entropy can serve as an excellent discriminator of likely patch regions. Importantly, we also show that differential entropy analysis detects effectively even when naturalistic adaptive attacks are applied; (2) More accurate localization of patches using a trained auto-encoder: we show that a primary reasons behind the limited effectiveness of prior solutions is their inaccurate localization of patches. We substantially improve patch localization, raising the rate of accurately located patches (IoU > 0.5) to twice compared to other related approaches. Moreover, Jedi improves the recovery rate while reducing by half the lost detection rate compared to the other evaluated adversarial patch defenses.

We conduct comprehensive experiments under different datasets and for different scenarios. Our results show that Jedi localizes adversarial patches with high precision, which consequently leads to an effective patch mitigation process, restoring up to 94% of incorrect results caused by adversarial patch attacks. More importantly, Jedi remains efficient with up to 76% recovery rate against GAN-based Naturalistic Patch [13], which almost completely bypasses other defenses. Besides, we propose a new adaptive attack that comprehensively limit entropy of the generated patch in Section 6. We find that it is difficult to limit adversarial noise entropy without losing the patch efficiency. As a result, we believe that entropy is a strong feature to discriminate between adversarial patches and benign images.

Our contributions can be summarized as follows:
i) We propose Jedi, an entropy-based approach to defend against adversarial patches. Jedi leverage an entropy analysis based approach for a precise patch localization, which results in a high recovery success using inpainting. To our knowledge, we are first to use entropy for adversarial patch localization.
ii) We evaluate our defense in a variety of settings for both classification and detection tasks and find that Jedi recovers up to 94% of successful adversarial patch attacks
iii) We propose an entropy-aware adaptive attack that considers entropy budget in the patch generation process. We show that limiting patch entropy escapes detection, but reduces considerably the patches' adversarial impact.
iv) For reproducible research, we open source our code (provided in supplementary material with a demo video).

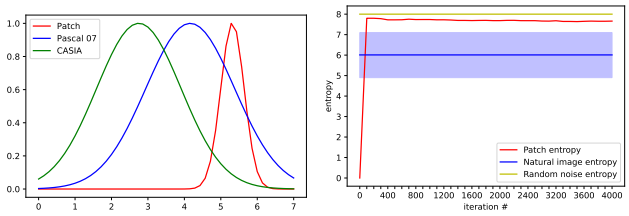
2. Preliminaries: Entropy Analysis

We provide a preliminary analysis from an information theory perspective to investigate entropy's discriminatory potential between natural images and adversarial patches.

Predictability in natural images. One measure of an image's *non-randomness* is the level of predictability: if one has access to parts of a given image, what is the capacity to guess the missing parts. If it is composed of totally random pixels, there is no predictability. However, natural images have semantic long-range correlations, and hence a high predictability is expected.

High entropy in adversarial patches. Adversarial patches are, by definition, not natural; they are the result of solving a constrained optimization problem. This adversarial perturbation is designed to be universal, which means that one specific noise is designed to fool a model for a variety of inputs. Moreover, the real-life settings of the threat model considered in adversarial patches represent additional spatial constraints on the adversarial noise generation. In fact, to build a plausible real-life attack, the noise has to be lim-

[1]We use the name Jedi, because the system recognizes chaos (high entropy patches) and restores peace by removing them, like the Jedis in the Star Wars movies.



(a) (b)

Figure 2. a) Comparison of entropy distributions of natural images from Pascal VOC 07 and CASIA datasets and adversarial patches. b)Evolution of patch entropy during the generation process

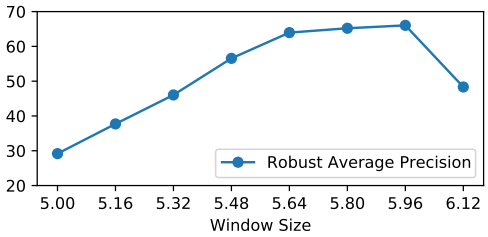


Figure 3. Robust Accuracy for different static entropy thresholds

ited by a specific location and cannot be distributed all over the input. Therefore, while natural images have a form of semantic continuity, adversarial patches concentrate a high amount of information within a restricted area, resulting in a highly *unpredictable* image.

Our intuition based on this observation is that adversarial patches should contain a statistically higher amount of information, from an information theory perspective, compared to any random neighborhood from a natural image distribution. Therefore, we use Shannon's entropy as an indicator of adversarial patch candidates. To verify this intuition, we propose a preliminary study to investigate the statistical components of different adversarial patches generated for several models. We also observe the patch entropy behavior in the adversarial noise generation process, comparatively with the mean entropy of the corresponding clean data distribution.

First, we propose to compare the statistical distribution of localized entropy levels in natural images and adversarial patches. The natural images used for this study come from cropping random 50×50 pixel areas from sample images in the datasets used for our experiments in Section 5. As for the adversarial patches, we create a collection of adversarial perturbations, as well as patches featured in other adversarial attack papers. Similarly to natural images, we use a sliding window to create 50×50 pixel sub-images.

Figure 2a shows a considerable entropy distribution shift with adversarial patches having approximately 30% higher mean entropy than natural images, at least. This shift is significant enough to be exploited as a metric to distinguish adversarial patches from natural images. However, the diverse environments and sparsity of natural images result in an entropy distribution with a large standard deviation, which in turn results in a slight overlap between both distributions. Therefore, a comprehensive entropy-based discrimination is required to avoid false positives.

To further explore the noise behavior at design time, we study the evolution of entropy levels in the adversarial patch during the patch generation process. We run the patch generation proposed in [3] while monitoring its entropy. The results are depicted in Figure 2b, along with: (1) the mean and standard deviation of the entropy distribution of natural images of the same size cropped from the Pascal VOC 07 dataset, and (2) the entropy of a totally random image representing the theoretical maximum entropy level for this signal window. The starting point for the patch generation process is a uniform image. With the noise exploration progress, the entropy quickly rises to exceed the mean entropy of similar size natural images, and is close to the theoretical maximum entropy. This suggests that effective patches trend towards higher entropy as their effectiveness rises, with entropy values being comparable to random noise.

3. Proposed Approach

The differential entropy analysis in Section 2 represents the basis of Jedi to locally discriminate adversarial patches from their surrounding natural image data. Our framework, as illustrated in Figure 1, is based on 3 main steps:

3.1. Locating high entropy kernels

Adversarial patches are likely located in the high entropy clusters. The first step of our approach is to identify these clusters by building a heat map of *local entropy*. A sliding window is applied through the image, where the local Shannon's entropy of the window is calculated. The resulting local entropy values form an entropy heat map. To keep only high entropy kernels in the heat map, a thresholding operation is applied.

Threshold: An appropriate threshold is key to properly isolating the high entropy kernels that constitute potential patch candidates. A too high threshold leads to the inability to detect adversarial patches, while a too low threshold results in a high false positive rate. As Figure 3 shows, the evolution of the robust accuracy as the entropy threshold lowers suggests that using a static threshold is not an appropriate choice. In fact, uncontrolled settings such as outdoor environments have significant variation, which makes a threshold for one image not necessarily fit for another image. Therefore, we propose to set a dynamic entropy

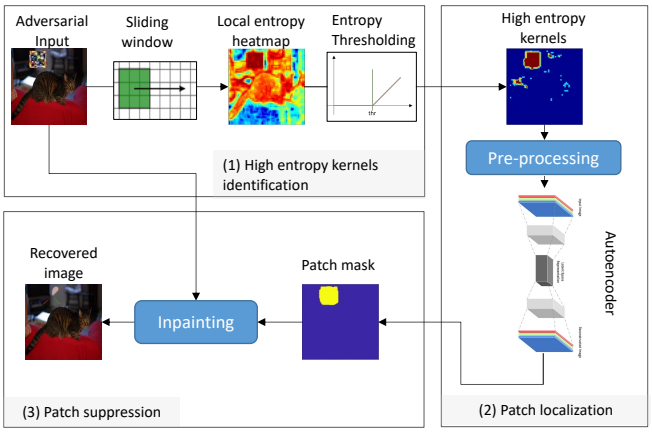


Figure 4. Detailed diagram of our Jedi adversarial patch defense

threshold defined by Equation 1:

$$thr = \mu_{clean} + (w_{tolerance} + w_{image})\sigma_{clean} \quad (1)$$

Where: μ_{clean} and σ_{clean} are, respectively, the mean and standard deviation of the clean samples entropy, $w_{tolerance}$ is an empirical weight to adjust the threshold according to the risk strategy, w_{image} is the weight used to adjust the threshold according to the entropy distribution of the current image. Finding the threshold requires a knowledge of the local entropy distributions: An entropy distribution of known clean images obtained from the considered dataset or application, and the entropy distribution of the current input image, easily extracted from the heat map.

Parameter exploration: The *local entropy* exploration is based on the neighborhood definition, which depends on empirical spatial parameters such as the window size and the padding. Therefore, we proceed to a space exploration to select these parameters driven by maximizing the performance, which can be found in supplementary material.

As shown in Figure 4, the outcome of this step is a heat map that contains high entropy "kernels" where potential patch candidates are detected. However, this kernel detection is not sufficient for the final mask localization as it may be incomplete and/or contain non-patch locations. Further processing is required to refine this map into the final mask.

3.2. Patch shape reconstruction using Auto-Encoders

The next step of building the mask to surgically localize adversarial patches is to expand the high entropy kernels to the full accurate shape of the patch using autoencoders (AEs) as these architectures are adapted to data reconstruction from incomplete inputs. The high entropy kernels obtained in the previous step may be incomplete or contain false positives; there may be some overlap between adversarial patch entropy values and some parts of the image with naturally high entropy. However, natural high entropy areas mostly correspond to edges, which results in scattered high entropy kernels. Therefore, to improve the quality of the final patch mask, we perform a pre-processing filtering where we remove all scattered clusters. We explored several AEs and chose the most effective architecture with the lowest latency. We use a sparse Autoencoder (SAE), which uses regularization to enforce sparsity. The proposed AE's hyperparameters are as follows: (i) one input layer which takes the entropy heat map with the image size, (ii) one hidden layer of 100 neurons, and (iii) an output layer corresponding to the mask that identifies the patch location within the image. We use a sparse AE with a sparsity proportion of 0.15 and sparsity regularization of 4. To train the AE, we simulate an attack using a subset of images where we place a patch on the image, and extract a mask representing the patch's coverage; neither the images nor the patch were used other than in training the AE. Specifically, we collect these masks and use them as the training set of patch shapes for the AE to learn how to refine patch localization and provide a mask. We generated the AE training data by creating masks of the same size as the expected input images containing the most likely positions of adversarial patches and their potential sizes and shapes. The trained AE uses the high entropy kernel map as an input and outputs a mask with the reconstructed patch location.

3.3. Inpainting

The last step of Jedi aims at recovering the prediction (detection/classification) lost due to the adversarial patch. Therefore, Jedi overwrites the localized patch with data sampled from the immediate neighborhood distribution using inpainting. Inpainting has been used by prior work [5, 11, 25] to mitigate adversarial patch attacks given an accurate mask. Since the goal of this step is to defuse the patch, not to restore the exact original input, it is not required to have a pixel-perfect recovery of the original image. In fact, a sampling from the same distribution is sufficient for the current state-of-the-art DNN models to achieve correct output. We use the coherence transport based inpainting method [2]; Using the input image and the mask obtained from the previous step, we substitute mask pixels (starting from the boundaries and moving inwards) with a weighted sum of the values of external pixels within a certain radius. The weights are determined using the values of the coherence vectors in the pixel neighborhood.

4. Experimental Methodology

4.1. Evaluation metrics

We use the classical computer vision metrics (Accuracy for classification, Average Precision for detection) to evaluate the effectiveness of the attacks and our proposed de-

fense's success in removing the patches. However, while these metrics can show a global view of the results of the patch attacks and defenses, they are not specific enough to evaluate and debug the defense mechanisms. For example, since *patch detection* is fundamental for an accurate recovery, this needs to be evaluated separately. Moreover, the robust accuracy alone gives a global overview but does not allow a close examination of the defense behavior. In fact, if a given defense results in partially degrading the baseline accuracy and overall recovering, it is important to evaluate more precisely the impact. Therefore, we propose to additionally consider the following set of metrics for a detailed analysis of the defense:

i) Patch Detection Rate: It represents the percentage of applied patches that have been detected by the defense with an Intersection-over-union value that exceeds 0.5. This metric enables a precise evaluation of the localization performance of a given defense.

ii) Recovery Rate: The recovery rate represents the percentage of outputs restored by the defense, *relative to the number of successful attacks.* It models precisely the intrinsic positive impact of the defense.

iii) Lost Prediction Rate: This metric models the performance degradation caused by the defense. It represents the percentage of negatively affected *correct predictions* normalized to the set where the patch initially failed.

4.2. Experimental setup

Benchmarks: We evaluate Jedi in a variety of environments commonly used in computer vision:

- *Classification tasks:* We use ImageNet [7] for the large amount and variety of classes it contains. We also use Pascal Dataset [8] for the variety of entropy conditions: from very low background entropy (example: clear sky) to very high (urban environments and forests).

- *Detection tasks:* for the two detection tasks, we use INRIA dataset [6] to test attacks in an outdoor high entropy uncontrolled environment, and CASIA datasets [26] to test detection in an indoor controlled environment. Other benchmarks could be found in the supplementary material.

Adversarial Patches: In this evaluation, we use four state-of-the-art adversarial patches: Adversarial Patch [3] and LAVAN [14] are used against the classification tasks. And the YOLO adversarial patch [21] and the Naturalistic Patch proposed [13] are chosen to attack the detection tasks.

Experiments: For a comprehensive evaluation, we chose combinations (model/dataset) *where the generated patches were damaging*, which corresponds to a stronger attacker. This corresponds to ImageNet with [3], using ResNet50, Pascal VOC 07 with [3], using Resnet50, CASIA with [21], using YOLO and INRIA with [21]. Other combinations are available in the supplementary material.

Comparison to the state-of-the-art: We compare Jedi against LGS [17] and Jujutsu [5] in all of the experiments we describe earlier. We also for the sake of illustration compare with 3 certified defenses, namely, Derandomized Smoothing [15], Smoothed ViT [19] and Patchguard [22].

Table 1. Patch localization performance

Dataset	**Jedi** (ours)	LGS	Jujutsu
Imagenet + [14]	87.20%	44.50%	10.85%
Pascal VOC 07 + [3]	90.71%	34.30%	19.22%
CASIA + [21]	93.49%	57.25%	N/A
INRIA + [13]	38.80%	73,11%	N/A

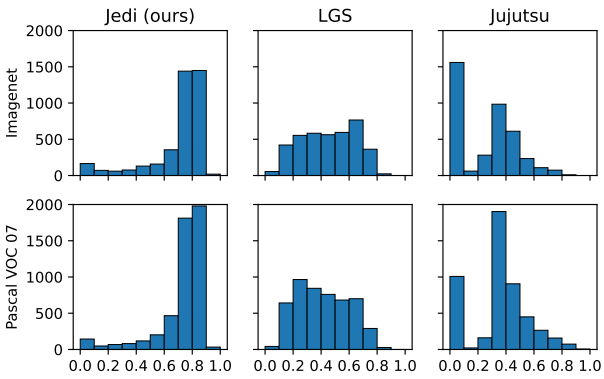


Figure 5. Comparison of detailed histograms of patch localization performance

5. Results

Patch localization performance – First, we focus on the patch localization performance; we measure the Intersection-over-Union (IoU) of the mask produced by Jedi, LGS and Jujutsu compared to the ground truth patch location and report the results in Table 1. For a comprehensive qualitative study of the patch localization, we provide the comparative distribution of the IoU in Figure 5, which depicts the occurrence of IoU scores in bins of 0.1 width. The two rows of the figure show the two sample experiments considered for this qualitative study , on Imagenet and Pascal VOC 07, while the columns compare three patch mitigation methods: Jedi (ours, left), LGS ([17],center) and Jujutsu ([5], right).

An efficient patch localization process is characterized by a maximum coverage of the patch pixels and minimum false positive pixels, leading to high IoU values. Therefore, the IoU distribution of an efficient patch localization method skew towards high IoU. As shown in Figure 5, Jedi has the highest shift to the right in the IoU distribution with more than 80% of IoUs are higher than 0.7, while less than 10% of the two related defenses have IoUs in this range.

End to end results – In Table 2, we report the *Clean Ac-*

Table 2. Clean, Adversarial and Robust accuracy of Jedi in classification benchmarks

Dataset	Clean Accuracy	Adversarial Accuracy	Robust Accuracy
Imagenet	74.10%	39.26%	64.34%
Pascal VOC 07	72.17%	26.94%	66.40%

Table 3. Clean, Adversarial and Robust accuracy of Jedi in detection benchmarks

Dataset	Clean Avg Precision	Adversarial Avg Precision	Robust Avg Precision
CASIA	91.47%	39.60%	88.21%
INRIA	53.22%	12.17%	28.03%

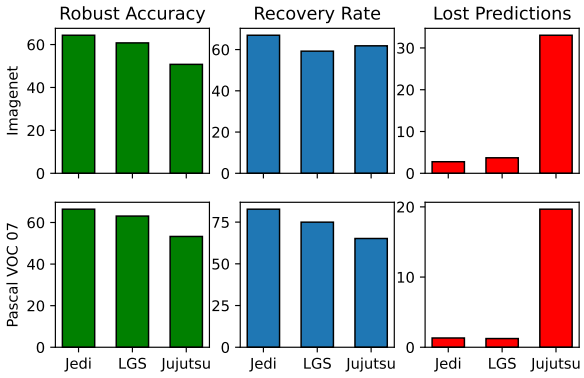


Figure 6. Comparison of Jedi's performance compared to other state of the art methods for classification tasks

curacy of the model prior to adding the adversarial attack, *Adversarial Accuracy* of the undefended model against the adversarial patch, and *Robust Accuracy* of the model after applying Jedi in the classification tasks. Similarly, we report Clean Average Precision, Adversarial Average Precision and Robust Average Precision for the detection tasks in Table 3. In Figure 6, we report the Recovery Rate, Lost Predictions and Accuracy for Jedi for each of the adversarial datasets for the classification tasks, and compare the results to LGS and Jujutsu, while Figure 7 depicts the comparative results in the detection tasks. Results show that Jedi outperforms related defenses for all metrics: Jedi restores most incorrect detections caused by adversarial patches while causing the lowest lost predictions. Further evaluation using different patch sizes is available in Section 6 of the supplementary material.

We also evaluate certified defenses on Imagenet dataset, using [3] and Resnet-50, the baseline results using an unprotected model are Clean Accuracy of 74.10%, the Patch Success rate is 49.02% and an Adversarial Accuracy of 39.26%. While a direct comparison is unfair, we wish to quantify the cost of ensuring provable robust defense on overall performance. As expected, their performance was lower than the empirical defenses, as shown in Table 4. These results show that while certified adversarial patch defenses offer provable robustness, their empirical utility is limited. These issues are more evident when the input images are larger or contain more aggressive attacks (such as bigger patches).

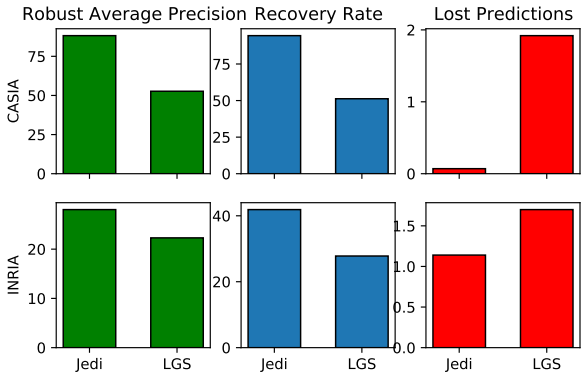


Figure 7. Comparison of Jedi's performance compared to other state of the art methods for detection tasks

Table 4. Performance of certified defenses on the Imagenet dataset

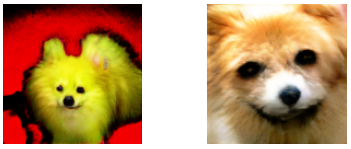
Defense	Jedi	[15]	[22]	[19]
Robust Accuracy	**64.34%**	35.02%	30.96%	40.38%
Recovery Rate	**66.95%**	27.65%	28.07%	36.34%
Lost Predictions	**2.77%**	52.56%	51.55%	33.54%

6. Adaptive Attacks

Empirical defenses against adversarial attacks have been shown vulnerable to adaptive attacks, where an attacker is aware of the defense and its parameters (i.e. a white box scenario). The adversary creates an adversarial patch that exploits specific weakness in the defense to bypass it. In this section, we investigate the robustness of Jedi to 2 adaptive attacks: (i) the first is a GAN-based naturalistic patch generation method, (ii) and the second is an **entropy-aware adaptive attack** that we propose.

6.1. Naturalistic Patch

A recent adversarial patch generation method that might be adaptive to our defense is the Naturalistic Patch [13]. We investigate the effectiveness of this attack since it generates stealthy adversarial patches that mimic real objects to avoid visual suspicion and therefore has the potential to evade detection. Samples of the naturalistic patch generated for our experiments are shown in Figure 8.



(a) CASIA dataset (b) INRIA dataset

Figure 8. Samples of the naturalistic patches

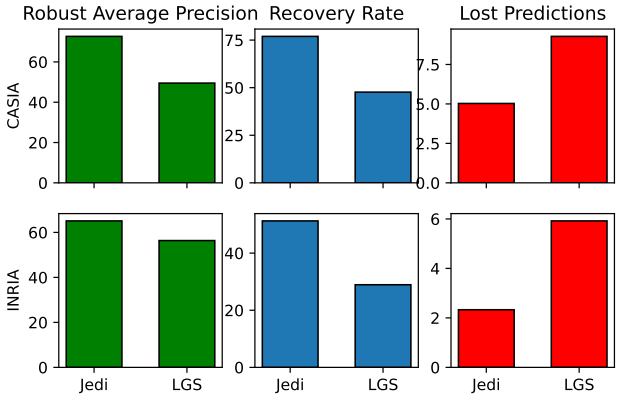


Figure 9. Comparison of Jedi's performance against the naturalistic patch

The assumption is that these naturalistic patches are outliers to our initial observation in terms of entropy distribution (Section 2). Therefore, one can expect these patches to evade our defense. We evaluate Jedi against the naturalistic patches and we summarize the results in Figure 9. Please note that we found that the naturalistic patch wasn't efficient in classification tasks (patch success rate didn't exceed 8%). For this reason, we show only detection benchmarks results. Surprisingly, the results indicate that Jedi is able to detect and mitigate the naturalistic patches with similar performance to regular patches, while LGS efficiency is considerably impacted by the Naturalistic Patch. The takeaway from this experiment is that adversarial patches, perhaps inherently, require high entropy even if they look naturalistic. In the following, we further investigate this hypothesis by generating an *entropy-bounded adversarial patch*.

6.2. Low-entropy adaptive patch

To create a patch that bypasses our defense, it should contain a low entropy by-design. Therefore, we propose an adversarial patch generation method with an objective to limit the patch entropy. However, since Shannon's entropy is not derivable we define a constraint on entropy instead of integrating it in the loss function. Specifically, we formulate the problem as a constrained optimization; Our objective is to fool a victim model $C(.)$ on almost all the input samples from a distribution μ in $\mathbb{R}^d$ given an entropy budget ε. This

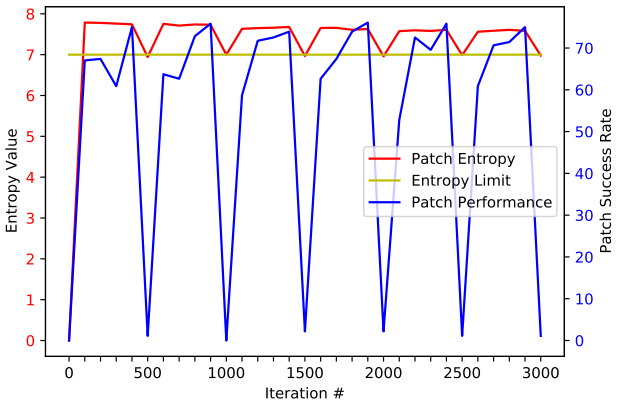


Figure 10. Entropy evolution during the adaptive patch generation

problem can be expressed as finding δ such that:

$$C(x + \mathcal{A}(\delta, k)) \neq C(x), \ for\ most\ x \sim \mu \\ s.t. \mathcal{H}(\delta) \leq \varepsilon \tag{2}$$

Where δ is the adversarial patch, which entropy is controlled by ε, $\mathcal{A}(.)$ is a function that applies a patch δ in a given location k within an input sample x.

The patch generation approach is shown in Algorithm 1; To enforce the entropy constraint, we need to project the noise δ back to the entropy-limited space. Therefore, we add a parallel process that halts the patch generation every n iterations to check whether the patch satisfies the entropy constraint. If not, a *local search for entropy reduction* is run to project the patch in the acceptable solutions space. This step is based on Variable Neighborhood Search, where for each iteration of entropy reduction, a search is initiated to find the best neighboring patch by replacing pixels of a certain color value by the closest (euclidean distance in the RGB space) color that already exists in the image. This step reduces the number of colors in the image, which in turn reduces the entropy. The selected neighbor is the patch that keeps the highest attack success rate. This process is repeated until patch entropy is below the budget, then the patch generation process resumes. Using this process, we attempt to create a patch that bypasses our defense on Pascal VOC 07 classification task. The results are shown in Figure 11 for patches with different entropy budgets, as well as a regular patch for reference. Figure 11 shows that the patches with a lower entropy limit can partially bypass Jedi, with lower patch detection rates and recovery rates. However, the low entropy patch has lost nearly all of its capabilities: The attack success rate drops from 64% down to only 8%. The source code as well as illustrations of the entropy limit's impact on the generated patches are available in the supplementary materials.

Algorithm 1 Low entropy patch generation algorithm

Input: n_{epochs}: number of training epochs ,I_{train} Training image set,$params$: Patch generation parameters, $checkFreq$: Frequency of entropy checks , ε: entropy limit
Output: Low entropy adversarial patch
for $e \in [1, n_{epochs}]$ **do**
  for $im \in I_{train}$ **do**
    δ = *PatchTraining*(params)
    if $it\%checkFreq == 0$ **then**
      */*Repeat each checkFreq iterations*/*
      patchEntropy $\leftarrow$ *Entropy*(δ)
      while $patchEntropy > \varepsilon$ **do**
        */*Target n random colors in the image for entropy reduction*/*
        colorList $\leftarrow$ *random*(0,255,n)
        $\delta \leftarrow$ reduceEntropy(δ,colorList)
      end while
      $patchEntropy \leftarrow$ *entropy*(δ)
    end if
  end for
end for

function *reduceEntropy*(δ, colorList)
*/*Find best patch among random color removal tries*/*
for $c \in colorList$ **do**
  */*Remove the targeted color and replace it with the most similar color found in the image*/*
  nearestColor $\leftarrow$ *findNearestColor*(c)
  newPatch[c] $\leftarrow$ *replaceColor*(δ, c, nearestColor)
  newPatchASR[c] $\leftarrow$ *testSuccessRate*(newPatch[c])
end for
*/*Keep the best performing patch */*
bestPatch $\leftarrow$ *argmax*(newPatchASR)
return bestPatch
end function

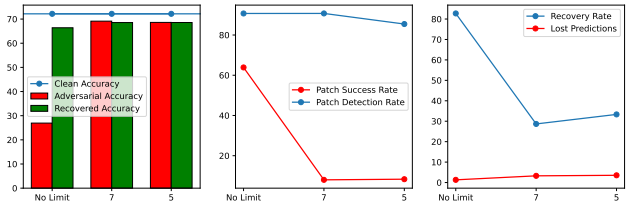


Figure 11. Performance of low entropy adaptive patches with comparison to a regular patch

7. Discussion and Concluding Remarks

In this paper, we present Jedi, a new defense against adversarial patches that is model agnostic, and robust against naturalistic patches. To our knowledge, this work is first to leverage a **differential entropy analysis** to detect adversarial patches. For a surgical localization, the entropy exploration outcome is fed to an AE which generates patches that are then overwritten by surrounding distribution through an inpainting. Jedi accurately localizes 85.3%-93.5% of adversarial patches across different datasets/attacks. Our qualitative analysis shows that the majority of detected patches have high IoUs. This confirms that *entropy distribution is a reliable metric to separate adversarial patches from benign images*. Given the efficient patch localization, the end-to-end post-inpainting results show accuracy/average precision restored up to their clean level. Across several benchmarks, Jedi recovers 67.0%-94.4% of the originally correct detections, with a low lost predictions compared to related work.

Comparison with certified defenses. To compare with certified defenses, we tested Patch Cleanser [23] comparatively to Jedi. For Imagenet benchmark with Resnet50, Patch Cleanser with 2x2 mask grid achieved 56.97% non certified robust accuracy, compared to 64.36% for Jedi. The certified accuracy for this benchmark is 4.87%. For this setting Jedi offers higher robustness under comparable time per frame. For a 6x6 mask grid, Patch Cleanser achieves 64.00% robust accuracy, but consumes $\sim 10\times$ more time than Jedi.

Adaptive attacks. We tested Jedi against an attack that uses GANs to generate naturalistic patches [13], which a priori seemed adaptive to our defense since it is based on distinguishing distribution of natural images from adversarial noise distribution. Interestingly, Jedi shows high robustness even against this attack. To further evaluate the vulnerability against potential adaptive attacks, we attempted to exploit what seems to be a weak spot for an adversary; A low entropy patch might go undetected by our defense. Our experiments show that to generate a Jedi-undetectable patch, the adversary has to sacrifice the attack efficiency, rendering the adaptive patch useless from an attacker perspective. On the other hand, rising the entropy budget increases the patch efficiency but accordingly makes it within Jedi detection capacity.

Limits. We believe that these adaptive attacks corroborate our initial intuition that adversarial patches might **inherently** require high entropy due to the information theoretical necessity of embedding too much information in a limited "channel". While this is based on empirical analysis, we believe that further analysis from information theoretical perspective is required to prove these findings.

Acknowledgment

This work has been supported in part by RESIST project funded by Région Hauts-de-France through STIMULE scheme (AR 21006614), and EdgeAI KDT-JU European project (101097300).

References

[1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[2] Folkmar Bornemann and Tom März. Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision*, 28(3):259–278, Jul 2007. 4

[3] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. 1, 3, 5, 6

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 1

[5] Zitao Chen, Pritam Dash, and Karthik Pattabiraman. Turning your strength against you: Detecting and mitigating robust and universal adversarial patch attacks, 2021. 1, 4, 5

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005. 5

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[8] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, Jun 2010. 5

[9] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 1

[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014. 1

[11] Jamie Hayes. On visible adversarial perturbations & digital watermarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 1, 4

[12] Shahar Hoory, Tzvika Shapira, Asaf Shabtai, and Yuval Elovici. Dynamic adversarial patch for evading object detection models, 2020. 1

[13] Yu-Chih-Tuan Hu, Bo-Han Kung, Daniel Stanley Tan, Jun-Cheng Chen, Kai-Lung Hua, and Wen-Huang Cheng. Naturalistic physical adversarial patch for object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7848–7857, October 2021. 1, 2, 5, 6, 8

[14] Danny Karmon, Daniel Zoran, and Yoav Goldberg. LaVAN: Localized and visible adversarial noise. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2507–2515. PMLR, 10–15 Jul 2018. 1, 5

[15] Alexander Levine and Soheil Feizi. (de)randomized smoothing for certifiable defense against patch attacks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6465–6475. Curran Associates, Inc., 2020. 1, 5, 6

[16] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors, 2018. 1

[17] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307, 2019. 1, 5

[18] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, 2016. 1

[19] Hadi Salman, Saachi Jain, Eric Wong, and Aleksander Madry. Certified patch robustness via smoothed vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15137–15147, June 2022. 5, 6

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013. 1

[21] Simen Thys, Wiebe Van Ranst, and Toon Goedeme. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1, 5

[22] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium (USENIX Security)*, 2021. 1, 5, 6

[23] Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. Patchcleanser: Certifiably robust defense against adversarial patches for any image classifier. In *USENIX Security Symposium (USENIX Security)*. USENIX Association, 2022. 8

[24] Chong Xiang and Prateek Mittal. Patchguard++: Efficient provable attack detection against adversarial patches. In *ICLR 2021 Workshop on Security and Safety in Machine Learning Systems*, 2021. 1

[25] Zirui Xu, Fuxun Yu, and Xiang Chen. Lance: A comprehensive and lightweight cnn defense methodology against physical adversarial attacks on embedded multimedia applications. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 470–475, 2020. 1, 4

[26] Shiqi Yu, Daoliang Tan, and Tieniu Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 441–444, 2006. 5