

ODDR: Outlier Detection & Dimension Reduction Based Defense Against Adversarial Patches

Nandish Chattopadhyay^{1*}, Amira Guesmi^{1*}, Muhammad Abdullah Hanif¹,
 Bassem Ouni², Muhammad Shafique¹

¹ eBrain Lab, Division of Engineering, New York University (NYU) Abu Dhabi, UAE

² Technology Innovation Institute (TII), Abu Dhabi, UAE

Abstract

Adversarial attacks present a significant challenge to the dependable deployment of machine learning models, with patch-based attacks being particularly potent. These attacks introduce adversarial perturbations in localized regions of an image, deceiving even well-trained models. In this paper, we propose Outlier Detection and Dimension Reduction (ODDR), a comprehensive defense strategy engineered to counteract patch-based adversarial attacks through advanced statistical methodologies. Our approach is based on the observation that input features corresponding to adversarial patches—whether naturalistic or synthetic—deviate from the intrinsic distribution of the remaining image data and can thus be identified as outliers. ODDR operates through a robust three-stage pipeline: Fragmentation, Segregation, and Neutralization. This model-agnostic framework is versatile, offering protection across various tasks, including image classification, object detection, and depth estimation, and is proved effective in both CNN-based and Transformer-based architectures. In the Fragmentation stage, image samples are divided into smaller segments, preparing them for the Segregation stage, where advanced outlier detection techniques isolate anomalous features linked to adversarial perturbations. The Neutralization stage then applies dimension reduction techniques to these outliers, effectively neutralizing the adversarial impact while preserving critical information for the machine learning task. Extensive evaluation on benchmark datasets against state-of-the-art adversarial patches underscores the efficacy of ODDR across diverse tasks and settings.

1. Introduction

Adversarial attacks represent a serious threat to the robustness and performance of deep neural network (DNN) models [7, 18]. These attacks involve the strategic introduction

of adversarial perturbations into test samples, leading to significant disruptions in the model’s predictions. A particularly effective form of these attacks is the insertion of localized patches into test images. This allows attackers to force DNN models into making errors in essential tasks such as image classification, object detection, and depth estimation [21, 25]. Patch-based attacks are a widely recognized and practical form of adversarial attack, particularly valued for their adaptability in scenarios with limited access to the system [20]. These attacks function much like discreet stickers, easily applied in real-world situations where adversaries may face constraints in resources or access. The subtlety of patch-based attacks makes them especially elusive, underscoring the critical need for robust defense mechanisms.



Figure 1. Demonstration of Adversarial Defense with ODDR [Task: Person Detection (Object Detection sub-task), Model: YOLOv4 [2], Patch: AdvYOLO [38], Dataset: INRIA [12]].

Standard defenses that work against imperceptible noise, including transformation-based defenses (JPEG compression, quantization, color compression [26], feature distillation [32] etc.) encounter a wall in the ability to provide robustness when it comes to the task of defending against adversarial patches, because these transformations unnecessarily affect the clean regions of the image sample as well and lead to a simultaneous loss in clean accuracy [8]. Therefore it is necessary to develop a scheme that identifies the region on the sample where the adversarial patch is present and thereby act specifically on that region. We address this

*These authors contributed equally to this work

by considering this as a two-part problem, one of identifying the location of the adversarial patch within the image, and the second problem being the ability to neutralize the adversarial behavior of the detected region. Our objective is to maximize the capabilities of both of these elements of adversarial defense (elaborated in Section 5.1), without reducing clean-case accuracy. Through ODDR, we propose utilizing the difference in distribution of the information within the adversarial patch and the rest of the image sample to identify the location of the patch. We achieve this separation using an outlier detection technique. ODDR is designed to minimize the rate of false positives during the detection of the region of the image containing the adversarial patch, through tunable parameters. For the second aspect of the problem, having identified the region that contains the adversarial patch, we advocate for the use of Dimension Reduction [15]. Through matrix-based transformations that help in projecting the sample into a lower dimensional space, this approach proves adept at neutralizing the impact of the patch without loss of surrounding pertinent information, as shown in Figure 1.

Novel Contributions – The main contributions are:

- We introduce a novel, model-agnostic defense method, Outlier Detection and Dimension Reduction (ODDR), which accurately identifies and localizes adversarial patches by detecting clusters of outliers within input features. By applying targeted dimension reduction over a window surrounding the detected outlier cluster, ODDR effectively neutralizes the adversarial patch while preserving crucial information.
- We demonstrate the efficacy and adaptability of ODDR across various tasks, including image classification, object detection, and monocular depth estimation. Our defense achieves a robust accuracy of 79.1% against the GoogleAP attack on the ImageNet dataset using CNN-based architectures and 83.1% using Transformer-based architectures.
- For object detection tasks, our defense mechanism achieves robust average precision scores of 93.54% and 83.21% when countering the AdvYOLO attack on the CASIA and INRIA datasets, respectively.
- In depth estimation, our defense significantly reduces the mean depth estimation error from 0.59 to 0.1 when mitigating the impact of SSAP-based attacks, demonstrating its effectiveness in preserving model accuracy.
- Our proposed approach improves the accuracy from 39.26% to 79.1% when performing GoogleAp attack, which surpasses that of the prominent state-of-the-art like LGS (53.86%), Jujutsu (60%), and Jedi (64.34%).

2. Theoretical Underpinning

In this section, we delve into the foundational theoretical principles that constitute the building blocks of the pro-

posed ODDR scheme. We outline the two components of the defense mechanism.

2.1. Segregating Input Features

We use an ensemble of decision trees to separate the outlier features from the rest of the image. Notably, this collection of decision trees, known as a Forest, can be tweaked to come up with an Isolation Forest algorithm, usable as a robust tool for detecting outliers and anomalies within datasets. Operating under the paradigm of Unsupervised Learning, this technique excels in scenarios where labeled data is unavailable, making it particularly versatile in diverse applications. It uses Binary Decision Trees through a bagging approach, drawing parallels to the well-known Random Forest technique employed in supervised learning. The core hypothesis guiding this technique is predicated on the assumption that anomalies represent a minority within the dataset, comprising fewer instances than their normal counterparts. Additionally, these anomalies are posited to possess attribute-values that diverge from those associated with normal instances. In simpler terms, anomalies are characterized as being both fewer in number and different from the rest of the samples [31]. This particularly makes it a good fit for detecting adversarial patches from input features of the test samples. Leveraging these assumptions, this technique excels in isolating anomalies by constructing an ensemble of Binary Trees for a given dataset. Anomalies, given their distinctive nature, tend to exhibit shorter paths within these trees compared to normal instances [22].

2.2. Outlier Detection

Consider a dataset $X = \{x_1, \dots, x_n\}$, comprising n vectors representing instances from a multivariate distribution of dimension d . To build the tree, the dataset X undergoes a recursive division process based on the random selection of an attribute q and a corresponding threshold value p . This process continues until one of the following three conditions is met:

- the decision tree attains maximum permissible height
- $|X| = 1$
- the data in the dataset X has equal values

The objective of anomaly or outlier detection can be achieved if one can deduce a way to rank the samples reflecting degree of anomaly. This can be done by running a sorting algorithm on the path lengths or anomaly scores. This naturally necessitates the definition of the anomaly scores. The path length $h(x)$ for a sample point x is defined as the count of nodes that x traverses in the tree from the root node to the conclusion of the traversal at an external node. Drawing on the structural similarities between isolation trees and Binary Search Trees, it can be inferred that the estimated average path length aligns with that of an unsuccessful search in a Binary Search Tree. For a dataset

comprising n samples, the average path length of an unsuccessful search is:

$$c(n) = 2H(n-1) - (2(n-1)/n), \quad (1)$$

with $H(i)$ denoting the Harmonic number, estimated as $\ln(i) + 0.57721567$ [Euler's constant]. Additionally, $c(n)$ represents the mean of $h(x)$ over n . Consequently, the anomaly score s can be defined as:

$$\text{Score}(x) = s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (2)$$

where $E(h(x))$ represents the mean of $h(x)$ derived from the forest, which is constructed from multiple such trees. It follows that when $E(h(x)) \rightarrow c(n)$, $s \rightarrow 0.5$; when $E(h(x)) \rightarrow 0$, $s \rightarrow 1$; and when $E(h(x)) \rightarrow n-1$, $s \rightarrow 0$. The interpretation of the anomaly score s is that if the samples possess an anomaly score close to unity, they are highly likely to be outliers and those with a score much smaller than 0.5 are likely to belong to the distribution. If a significant portion of samples returns a score around the 0.5 mark, it is highly probable that there are not many outliers in the dataset. Anomalies are identified by their shorter path lengths within the tree structure, with path length denoting the number of edges traversed from the root node.

2.3. Dimension Reduction

Dimension Reduction is typically used in feature engineering tasks in statistical learning, but this can also be used as an adversarial defense here, as explained below. This technique maps the input into a lower dimensional space, preserving a chosen amount of variability, thereby getting rid of significant proportion of the adversarial noise [4, 5]. Intuitively, this projects the matrix on to an orthonormal basis and the variability within it is expressed as a linear combination of the components [15]. At first, if we consider that an image (a colored image has three channels and therefore three such matrices) is a $m \times n$ -matrix M , then one can use Singular Value Decomposition to factorize the same into $M = U\Sigma V^T$, where U and V are orthogonal matrices and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$, where $r = \min(m, n)$ such that $\sigma_1 \geq \dots \geq \sigma_r \geq 0$. In this case, the singular values are the σ_i s, and thereby we have the top r columns of V and U being the right and left singular vectors respectively. This formulation is necessary for dimensionality reduction using Truncated Singular Value Decomposition as explained hereafter. For the matrix M with a rank r , one can represent $M = U_r \Sigma_r V_r^T$ where U_r and V_r comprises of the left and right singular vectors. For any k , such that $(k < r)$, one can therefore obtain M_k , where $M_k = U_k \Sigma_k V_k^T$, which is lower rank approximation of the matrix. This is used in dimensionality reduction, facilitated by the fact that the singular values are so arranged such that they provide the ordering of the information contained in such truncated approximations [16]. Specifically, in our implementation, we have

used the proportion of the singular values as a surrogate to estimate the amount of information preserved upon the image mask M being projected to a lower dimension. We have denoted this information preservation parameter as inf and used it as a tuning parameter for our defense mechanism.

3. ODDR Defense Mechanism

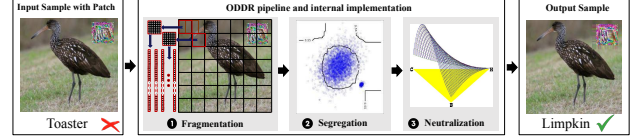


Figure 2. Overview of the Proposed ODDR Defense Methodology: The three-stage pipeline—Fragmentation, Segregation, and Neutralization—demonstrating the process of identifying and mitigating adversarial patches in the input image features. [Task: Image Classification, Model: [24], Patch: Google Adversarial Patch [3], Dataset: imagenet [13]]

As illustrated in Figure 2, the proposed ODDR defense technique encompasses three stages. The initial stage involves splitting the image into kernels using a moving window. This fragmentation process deconstructs the image into equally sized chunks, the dimensions of which are fine-tuned as hyper-parameters for specific applications. This stage is referred to as Fragmentation. Subsequently, the obtained fragments are flattened into vectors, and the outlier detection task is executed on these vectors, denoted as Feature Segregation. Finally, Dimension Reduction is applied to a window positioned on the cluster of fragments identified as outliers. This step aims to neutralize the adversarial patch without compromising information in the surrounding vicinity. The entire algorithm is detailed in Algorithm 1, with each component and function elucidated in the subsequent subsections.

3.1. Fragmentation

The initial step in processing the input image is Fragmentation. This involves segmenting the image into partially overlapping fragments using a moving square kernel. The size of this kernel, along with the stride length determining the movement for each new fragment, is a system hyper-parameter fine-tuned for optimal performance. Upon generating the fragments, each with three channels and a square shape, the data is flattened into long vectors. This flattening is a crucial preparatory step for the subsequent stage in the process, Feature Segregation.

3.2. Segregation

The notion of segregation of anomalous input features within the feature space is carried out as an outlier detection exercise. We propose that an ensemble of decision trees,

Algorithm 1 ODDR: Outlier Detection & Dimension Reduction

IN : I : original image, k : kernel size, str : stride length, c : confidence of anomalies, T : number of trees in the forest, d : size of square mask, inf : information preserved after SVD, $minPts$: minimum fragments in an anomaly cluster
OUT : I' : Image with neutralised patch

*/*Fragmentation*/*
Generate n fragments $X \leftarrow (x_1, \dots, x_n)$ from image I , using kernel size = k and stride = str

*/*Segregation*/*
Set $s = 0.3 \times size(X)$
 $iForest = Forest(X, T, s)$ (From Algo 2)
initialize $Y = (y_1, \dots, y_n)$
for $i = 1$ to n **do**:
 $y_i = Score(x_i)$, where $x_i \in X$ (From Eq. 2)
end for
Sort Y in descending order
Select $(1 - c) * n$ top y_i s as anomalies $(z_1, \dots, z_r) \rightarrow Z$
Select clusters $(C_1, \dots, C_p)$ in Z such that $size(C_i) \geq minPts$

*/*Neutralization*/*
Calculate cluster centers $z_cen_{C_i}$ and masks M_{C_i} of size d as:
for $i = 1$ to p **do**:
 Cluster center $z_cen_{C_i} = max(z_i) \forall z_i \in C_i$
 Create masks M_{C_i} from $loc(z_cen_{C_i} - (d/2))$ to $loc(z_cen_{C_i} + (d/2))$
 Perform Dimension Reduction using Singular Value Decomposition as:
 for $i = 1$ to p **do**:
 $M'_{C_i} \leftarrow SVD(M_{C_i}, inf)$
 Superimpose M'_{C_i} in place of M_{C_i} in image I to get I'
return I'

Algorithm 2 $Forest(X, T, size)$

IN: $X = (x_1, \dots, x_n)$: input data, T : number of trees in the forest, s : sample size
OUT: a $Forest$ ($Trees$ ensemble)
initialize : $Forest$
set $l_{max} = ceiling(log_2 s)$
for $i = 1$ to T **do**:
 $X' \leftarrow Sample(X, s)$
 $Forest \leftarrow Forest \cup Tree(X', 0, l_{max})$ (From Algo 3)
end for
return $Forest$

Algorithm 3 $Tree(X, h, l_{max})$

IN: $X = (x_1, \dots, x_n)$: input data, h : height of tree, l_{max} : maximum height of tree, Q : attributes of X
OUT: a $Tree$
if $h \geq l_{max}$ or $|X| \leq 1$, **then**:
 return $externalNode\{Size \leftarrow |X|\}$
else
 $X_{left} \leftarrow Select(X, q < p)$, where $min(q) \leq p < max(q)$, $q \in Q$
 $X_{right} \leftarrow Select(X, q \geq p)$, where $min(q) \leq p < max(q)$, $q \in Q$
 return $internalNode$
 {
 $LeftTree \leftarrow Tree(X_{left}, h + 1, l_{max})$
 $RightTree \leftarrow Tree(X_{right}, h + 1, l_{max})$
 $attribute \leftarrow q$
 $value \leftarrow p$
 }
endif

which is a forest, is best suited for this purpose. This algorithm is inspired by the Isolation Forest algorithm. The fragments generated using the aforementioned Fragmentation process serves as the dataset for the outlier detection algorithm. The anomaly detection using the Forest is outlined in Algorithm 2, which is essentially an ensemble of decision Trees which are generated using Algorithm 3. Once the trees are generated, the path lengths for the leaf

nodes are calculated using the Equation 1, which is thereafter used to estimate the anomaly score using the Equation 2, as described earlier. Once the anomaly scores (y_i) are ascertained to each fragment (x_i), the top outliers are selected (Z) based on a pre-defined threshold (c), which is set as a hyper-parameter. Therefore, Z contains fragments z_i s, detected as outliers. Since Fragmentation is a deterministic process, using k and str , we know the placement of every fragment x_i , and so for every $z_i \in Z$, we check the overlapping fragments to look for the clusters (C_i s), such that the size of each cluster is greater than a pre-defined hyper-parameter $minPts$.

3.3. Neutralization

For every detected cluster $(C_1, \dots, C_p)$, the cluster center of each is ascertained to be the fragment with the highest anomaly score. Considering the central pixel of that particular fragment as the center, the square masks are generated of size d , extending from the location $[z_cen_{C_i} - (d/2)]$ to $[z_cen_{C_i} + (d/2)]$ on either side of the central point. The mask derived from the Segregation process encompasses the entirety or, at the very least, the majority of the adversarial patch, whether naturalistic or otherwise. This assertion is grounded in the fact that the mask is chosen from the region where the cluster of anomalies or outlier fragments is situated. *Comprehensive testing and validation of this approach are presented in the Experiments Section later in the paper.*

To render the detected adversarial patch ineffective, two techniques have been explored. The first approach, though sub-optimal, involves replacing each pixel in the mask with the mean value of all the pixels within the region covered by the mask. This method performs effectively when the mask aligns closely with the adversarial patch. However, for a broader region, it may result in a potential loss of valuable information in areas where the mask does not completely overlap with the adversarial patch. To solve this problem, we propose Dimension Reduction. We apply a singular value decomposition on the mask, with a specific hyper-parameter that selects the proportion of information to preserve (corresponding to the number of singular values).

4. Experimental Results

In this section, we conduct a thorough evaluation of the effectiveness of our proposed defense mechanism against various patch-based attacks across different classification, object detection, and depth estimation tasks.

4.1. Experimental Setup

We assess ODDR performance across diverse settings commonly used in computer vision applications:

Task: Classification -

Datasets: We use the ImageNet dataset [13] for its extensive class variety and the Caltech-101 dataset [29] for specific object recognition tasks.

Models: Our assessment uses ResNet-152 [24], ResNet-50 [24], VGG-19 [35], Inception V3 [39], Vision Transformers [42] and Swin Transformers [33] for ImageNet and Caltech-101 classification tasks.

Task: Detection -

Datasets: For outdoor detection challenges, we use the INRIA dataset [12], capturing uncontrolled environmental variations and includes instances of multiple patches. For indoor scenarios, we use the CASIA dataset [41].

Models: The YOLO model [2] is employed for detection tasks. We test both outdoor and indoor scenarios.

Task: Monocular Depth Estimation (MDE) -

Datasets: For monocular depth estimation challenges, we use the KITTI dataset [17]. The dataset encompasses a wide variety of road types, including local streets, rural roads, and highways, providing a robust and diverse set of scenarios for depth estimation tasks.

Models: We employ the MIMdepth model [9] for monocular depth estimation tasks. MIMdepth utilizes a self-supervised learning approach, combining masked image modeling (MIM) with a transformer architecture to enhance the accuracy and robustness of monocular depth estimation.

Threat Model: We operate under the strongest assumption, a white-box scenario, where the attacker possesses complete knowledge of the victim DNN, including its architecture and parameters. As in other proposed defenses [28, 34, 40], the adversary is capable of substituting a specific region of an image with an adversarial patch, with the goal of consistently inducing misclassification/misclassification across all input instances.

Patch-based Attacks: In this evaluation, we employ seven state-of-the-art adversarial patches to rigorously assess model performance. For classification tasks, we utilize the Adversarial Patch (GAP) [3], LAVAN [27], Generative Dynamic Patch Attack (GDPA) [30], and Shape Matters (SM) [10]. For detection tasks, we apply the AdvYOLO adversarial patch [38] and the Naturalistic Patch [25]. Additionally, to evaluate Transformer-based monocular depth estimation (MDE) models, we use the shape-sensitive adversarial patch (SSAP) [21]. Further details in the Appendix. These patches represent cutting-edge techniques in the domain, ensuring a robust and comprehensive assessment of the ODDR defense mechanism across different challenges.

4.2. Results

Throughout the experiments, for our defense, we used a kernel size of 20x20 pixels, with a stride length of 10 for Fragmentation. For Segregation we varied the confidence of anomalies c between 0.8 – 0.85 and used 500 trees in the isolation forest. For the Dimension Reduction using SVD,

the information preservation parameter is varied between 70 – 85%.

4.2.1. ODDR in Classification Benchmarks:

In our evaluation, the primary focus was on assessing the model’s robust accuracy under adversarial conditions. To demonstrate the effectiveness of our defense strategy, we first generated adversarial patches using four distinct attack methods: LAVAN, GoogleAP, GDPA, and Shape Matters (SM). We then measured the model’s robust accuracy across varying patch sizes and architectures, including both CNN and transformer-based models. Table 2 showcases the outstanding performance of our defense technique, which significantly enhances robust accuracy against the aforementioned attacks on the ImageNet and CalTech datasets, with minimal impact on baseline accuracy. Notably, our defense achieves robust accuracy rates of 79.1% against GoogleAP and 74.1% against LAVAN, underscoring its effectiveness in mitigating these adversarial threats.

IMAGENET Model	Clean	GAP		LAVAN		GDPA		SM	
		Adv	ODDR	Adv	ODDR	Adv	ODDR	Adv	ODDR
ResNet 152	81.2	39.3	79.1	10.1	74.1	56.4	73.1	15.1	78.3
ResNet 50	78.4	38.8	75.6	10.2	70.2	55.8	72.6	16.4	72.8
VGG 19	74.2	39.1	72.8	11.1	71.1	60.1	71.8	17.1	71.9
Inception v3	79.8	38.2	77.6	10.9	77.1	62.1	72.9	12.8	74.1
ViT_b.16	85.8	36.7	83.1	9.2	82.9	68.4	82.6	11.9	82.3
swin_v2.b	84.5	35.9	81.9	6.9	81.2	69.8	81.8	13.4	80.8

CALTECH-101 Model	Clean	GAP		LAVAN		GDPA		SM	
		Adv	ODDR	Adv	ODDR	Adv	ODDR	Adv	ODDR
ResNet 152	94.1	48.6	90.8	15.6	91.1	45.7	88.4	23.6	91.8
ResNet 50	90.9	49.2	86.4	17.1	87.3	47.9	87.1	25.9	86.3
VGG 19	88.6	47.1	85.6	15.3	84.9	46.1	85.2	21.4	86.6
Inception v3	89.4	48.1	86.5	18.2	85.4	49.2	86.8	19.4	84.1
ViT_b.16	95.3	45.2	91.8	13.8	92.1	51.2	91.9	25.1	92.3
swin_v2.b	94.9	46.3	90.7	17.9	90.3	48.3	92.1	20.8	92.7

Table 1. ODDR robustness for Image Classification task.

Effectiveness against Subtle Patches: We test ODDR against subtle/invisible and inconspicuous patches which include IAP [1] and DorPatch [23], for further analysis.

Model	Attack	Clean	Adv	ODDR
ResNet-50	IAP	78.4	37.1	63.1
	DorPatch	78.4	4.5	59.8
Inception-v3	IAP	79.9	38.2	62.8
	DorPatch	79.9	7.3	61.7

Table 2. ODDR robustness for Image Classification task on ImageNet dataset against subtle attacks [using $k = 20$, $d = 20$].

4.2.2. ODDR in Detection Benchmarks:

In the context of detection tasks, we evaluated the Robust Average Precision and Recovery Rate against the AdvYOLO attack on the INRIA and CASIA datasets, as presented in Table 3. The Recovery Rate measures the proportion of successfully recovered outputs by the defense relative to the total number of successful attacks, providing a clear indication of the defense mechanism’s effectiveness. Our defense

technique demonstrates exceptional performance, achieving a robust average precision of 93.54% on the INRIA dataset and 83.21% on the CASIA dataset.

Dataset	Clean Accuracy	Adversarial Accuracy	Robust Average Precision
INRIA	96.42%	39.28%	83.21%
CASIA	96.77%	32.25%	93.54%

Table 3. Performance of ODDR against AdvYOLO-based attack.

4.2.3. ODDR in Depth Estimation Benchmarks:

For depth estimation tasks, we assessed the effectiveness of our defense using three key metrics, as defined in [21]: mean depth estimation error (E_d), the ratio of the affected region (R_a), and mean square error (MSE). Further details on these metrics can be found in the Appendix. As demonstrated in Table 4, our ODDR technique significantly mitigates the impact of the SSAP patches, reducing the mean depth estimation error from 0.59 to 0.1 and shrinking the affected region from 99% to 21%, thereby effectively preserving the accuracy of depth estimation.

KITTI	E_d	R_a	MSE
SSAP	0.59	0.99	0.53
ODDR	0.11	0.21	0.04

Table 4. Performance of ODDR against SSAP-based attack.

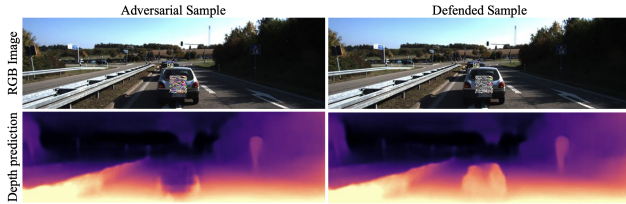


Figure 3. Illustration on the impact of ODDR on depth SSAP-based attack [21].

4.3. ODDR vs State-of-the-art

We conducted a comprehensive comparative analysis of ODDR against several state-of-the-art defenses, including LGS [34], Jujutsu [11], and Jedi [37], across all previously outlined experiments. For further context, we also compared ODDR with two certified defenses: De-randomized Smoothing [28] and PatchGuard [40]. As shown in Table 5, our defense technique outperforms existing methods, achieving a robust accuracy of 79.1%. This surpasses the robust accuracy achieved by LGS (53.86%), Jujutsu (60%), and Jedi (64.34%). Additionally, we compared ODDR with a Dimension Reduction-based defense [6], which achieved an accuracy of 66.2%, further highlighting the superior performance of our approach.

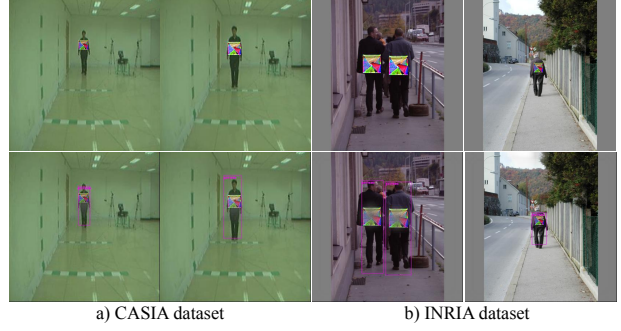


Figure 4. Illustration of the impact of ODDR on AdvYOLO patch-based attacks [38]: *Top row*: Adversarial images. *Bottom row*: Defended images for two datasets: a) CASIA representing indoor settings, and b) INRIA representing outdoor settings.

In addition, we compared our defense with two certified defenses to quantify the trade-off between ensuring provable robustness and overall performance, despite the inherent unfairness of this comparison. As expected, certified adversarial patch defenses demonstrated lower performance compared to empirical defenses, with De-randomized Smoothing [28] achieving 35.02% robust accuracy and PatchGuard [40] achieving 30.96% robust accuracy.

Model	Attack	Clean	Adv	LGS	Jedi	DR	ODDR
ResNet 152	GAP	81.2	39.3	53.9	64.4	66.2	79.1
	LAVAN	81.2	10.1	49.9	54.6	59.4	74.1
	GDPA	81.2	56.4	64.2	61.5	60.2	73.1
Inception v3	GAP	79.8	38.2	55.1	66.8	61.3	77.6
	LAVAN	79.8	10.9	50.7	59.4	55.4	77.1
	GDPA	79.8	62.1	66.9	71.4	58.6	72.9
swin_v2.b	GAP	84.5	35.9	65.2	70.6	62.6	81.9
	LAVAN	84.5	6.9	60.9	57.8	53.8	81.2
	GDPA	84.5	69.8	76.3	68.4	58.9	81.8

Table 5. ODDR comparison across attacks on ImageNet dataset.

In detection tasks, as demonstrated in Table 6, our defense outperforms in terms of both robust average precision and recovery rate when deployed against the advYOLO patch, across both INRIA and CASIA datasets.

Defense	INRIA		CASIA	
	Robust Avg. Precision	Recovery Rate	Robust Avg. Precision	Recovery Rate
LGS [34]	21%	27%	84%	48%
Jedi [37]	28.03%	42%	80%	50%
Ours (ODDR)	82.14%	43.86%	93.54%	61.29%

Table 6. Performance of our proposed defense ODDR compared to two state-of-the-art defenses against AdvYOLO.

5. Discussion

In this section, we explore additional performance aspects of ODDR, including its resilience against adaptive attacks, results from the ablation study, impact on model interpretability, computational cost, and hyperparameter tuning.

5.1. Ablation Study

The proposed ODDR technique identifies the adversarial noise in the patches as an outlier to the data distribution of the rest of the input sample. For each component, we chose the best algorithm that maximizes the overall inference accuracy. For *Outlier Detection*, we found that the modified Isolation Forests outperform clustering based techniques like DBSCAN [14]. The percentage gain in the overlap of the actual adversarial patch and the detected outliers cluster was up to 20% (ImageNet). For *Neutralization*, we used naive techniques like replacement of all pixels in the mask with the lowest value, highest value and the average value, and Dimension Reduction significantly outperformed them in the downstream machine learning task, by up to 15% (ImageNet) and is able to match that of Vision Transformers with significantly lower computations. This can be attributed to the fact that Dimension Reduction carefully preserves a certain amount of information in the mask, and is specifically helpful when there are false positives generated upstream. Notably, ODDR provides a performance gain of up to 13% over just Dimension Reduction [6], which can be primarily attributed to the Outlier Detection operation.

5.2. Impact of ODDR on Model Interpretability

We conduct a detailed visualization of the Grad-CAM [36] output for the adversarial example, both before and after the use of our defense. Notably, as shown in Figure 5, we observe a significant shift in the network’s focus when the adversarial patch is applied, as the attention shifts from the actual object to the manipulated patch. However, upon applying our defense, we observe a restoration of the network’s attention back to the original object, highlighting the efficacy of our defense in preserving the model interpretability.

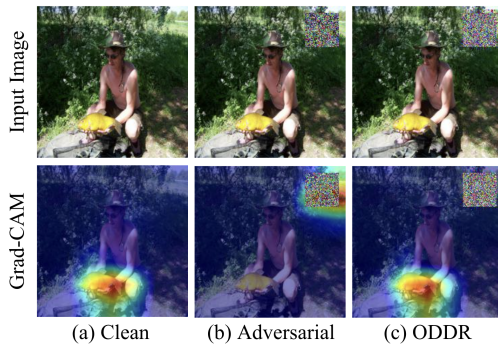


Figure 5. Grad-CAM visualization result for ODDR in action.

5.3. Computational Cost

Our defense mechanism, ODDR, is designed with low computational overhead, making it suitable for deployment on edge devices without requiring GPUs. The computational complexity of the Decision Forest is of the order $O(n * t * \log s)$ where s is the sub-sample size, n is the number of test samples, and t is the number of trees. The Truncated SVD has a complexity of $O(d^3)$ where d represents the dimensionality, which is typically small as it correlates with the size of the detected outlier clusters. In practical terms, the average processing time per sample is as follows: 0.034 seconds for ImageNet and CalTech-101 datasets, and 0.041 seconds per frame for the INRIA and CASIA datasets, when executed on an AMD Ryzen 5955 CPU. This is compared to the baseline processing time of 0.019 seconds on NVIDIA GPUs using ResNet50.

5.4. Hyper-parameter Tuning

ODDR uses two categories of hyper-parameters: *Active hyper-parameters* and *Passive hyper-parameters*. The active hyper-parameters significantly influence the performance of ODDR, and the results reported in the tables correspond to their optimal tuned values. These include c (the confidence level for identifying anomalies, ranging from 0.8 to 0.95), and inf (the information preserved after SVD, ranging from 0.7 to 0.9). The passive hyper-parameters, while less impactful on performance, were kept consistent across most experiments. These include k (kernel size, between 20 and 40 pixels), str (stride length, set to $k/2$, T (the number of trees in the Decision Forest, fixed at 100), the sampling fraction for the Trees (30%), and the minimum number of fragments required to form an outlier cluster (20-30) and d (the size of the mask applied to outlier fragments, set at 50 pixels). Additional details on these parameters can be found in the Appendix.

5.5. ODDR vs. Naturalistic Patches

A recent adversarial patch generation method that poses a potentially greater challenge to our defense is the Naturalistic Patch [19, 25]. We investigate the effectiveness of these attack strategies, considering its ability to generate stealthy adversarial patches that mimic real objects, rendering them less visually suspicious and, consequently, more challenging to detect. The results, specifically the recovery rate upon using ODDR, are presented in Tables 7. Despite the naturalistic nature of this patch and its seamless integration into the image, our defense successfully mitigated its adversarial impact. As an illustration, our defense demonstrates a notably robust average precision, surpassing that of previously proposed defenses such as LGS and Jedi. Specifically, our defense achieves a 65% higher average precision compared to LGS, and 55% and 63% for the INRIA dataset, respectively.

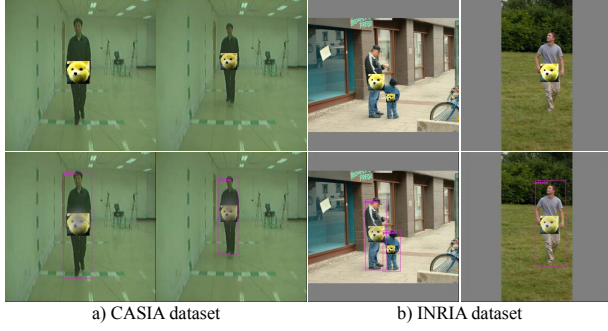


Figure 6. Illustration of the impact of ODDR on Naturalistic patch-based attacks [25]: *Top row*: Adversarial images. *Bottom row*: Defended images for two datasets: a) CASIA representing indoor settings, and b) INRIA representing outdoor settings.

Defense	INRIA		CASIA	
	Robust Avg. Precision	Recovery Rate	Robust Avg. Precision	Recovery Rate
LGS [34]	55%	29%	50%	48%
Jedi [37]	63%	51%	67%	76%
Ours (ODDR)	65%	47%	87.5%	90.27%

Table 7. Performance of our proposed defense compared to two state-of-the-art defenses against Naturalistic Patch [25] attack for INRIA and CASIA dataset.

5.6. Adaptive Attack

Adaptive attacks exploit full knowledge of the defense mechanism to bypass it. Since decision trees and Isolation Forests are non-differentiable, their direct integration into standard attack optimization processes is not possible.

5.6.1. Distribution Mapping

The detection of the location of the adversarial patch relies on the observation that the distribution of the information within the patch is different from that in the rest of the sample. To circumvent this, we map the adversarial patch’s variability to that of randomly selected image fragments across all channels, fitting Gaussian distributions to each fragment. We constrain the patch’s updates so its distribution parameters (μ and σ) align with the averaged fragment parameters (m and std), within tolerances $0 \leq |\mu - m| \leq 0.5$ and $1 \leq \sigma/std \leq 2$ (dataset specific). Beyond these ranges, the Gaussian distributions are statistically separable. On ResNet50 with ImageNet, this approach reduces clean accuracy from 78% to 73%, with a robust accuracy of 76%.

5.6.2. Zeroth Order Gradient Attack

Since the decision-tree-ensemble method used in the detection of the outlier fragments is non-differentiable, we used a Zeroth Order Gradient Based Attack to estimate the gradients and craft the adversarial perturbation within the patch. We tested this on the ImageNet [13] dataset with the follow-

ing parameters:

- Patch size varied from 37x37 pixels to 50x50 pixels
- The perturbation parameter δ was varied between $0.001 \leq \delta \leq 0.2$

We have observed that the adversarial accuracy is well within 5 – 10% range of the clean case accuracy for the $0.001 \leq \delta \leq 0.2$ setting, which means that the attack is not strong enough. For all $\delta > 0.2$, the adversarial noise is easily detected by the defense mechanism as outlier fragments. This observation has been consistent across the different patch sizes.

5.6.3. Lower Dimensional Patch

The second component of the ODDR defense mechanism is to render the detected patch ineffective using Dimension Reduction. So, we considered an adaptive attack that restricts the adversarial patch to a lower dimensional space to begin with, using the exact same Singular Value Decomposition on the patch during training time (for both Google Adversarial patch [3] and LAVAN [27]). We have observed that even a 1% drop in the informational content using SVD, preserving 99% of the variability, is sufficient to render the patch ineffective. Further explanations in Appendix.

6. Related Works

There are two primary approaches to defend against patch-based attacks [28, 34, 40] :

Empirical defenses include methods like Localized Gradient Smoothing (LGS) [34], Jujutsu [11] and Jedi [37]. **Certified defenses** aim to provide provable robustness against attacks. For instance, De-randomized Smoothing (DS) [28], PatchGuard (PG) [40] etc.

7. Conclusion

We propose ODDR, specifically designed to counteract the formidable threat posed by patch-based adversarial attacks on machine learning models. This technique leverages a three-stage pipeline—Fragmentation, Segregation, and Neutralization — which effectively identifies and mitigates adversarial patches while preserving crucial information for machine learning tasks. Through rigorous experimentation on benchmark datasets and state-of-the-art adversarial patches, we substantiate its effectiveness on robustness on multiple machine learning tasks, whilst demonstrating minimal loss in clean accuracies.

Acknowledgment

This research was partially funded by Technology Innovation Institute (TII) under the ”CASTLE: Cross-Layer Security for Machine Learning Systems IoT” project and by the NYUAD Center for Cyber Security (CCS), funded by Tamkeen under the NYUAD Research Institute Award G1104.

References

- [1] Tao Bai, Jinqi Luo, and Jun Zhao. Inconspicuous adversarial patches for fooling image-recognition systems on mobile devices. *IEEE Internet of Things Journal*, 9(12):9515–9524, 2021. 5
- [2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 1, 5
- [3] Tom Brown. Adversarial patch. 2017. 3, 5, 8
- [4] Nandish Chattopadhyay, Anupam Chattopadhyay, Sourav Sen Gupta, and Michael Kasper. Curse of dimensionality in adversarial examples. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 3
- [5] Nandish Chattopadhyay, Subhrojyoti Chatterjee, and Anupam Chattopadhyay. Robustness against adversarial attacks using dimensionality. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 226–241. Springer, 2021. 3
- [6] Nandish Chattopadhyay, Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, and Muhammad Shafique. Defensedr: Defending against adversarial patches using dimensionality reduction. *arXiv preprint arXiv:2311.12211*, 2023. 6, 7
- [7] Nandish Chattopadhyay, Lionell Yip En Zhi, Bryan Tan Xing, Anupam Chattopadhyay, and Muhammad Shafique. Investigating spatially correlated patterns in adversarial images. In *2024 IEEE International Conference on Image Processing Challenges and Workshops (ICIPCW)*, pages 4058–4064. IEEE, 2024. 1
- [8] Nandish Chattopadhyay, Abdul Basit, Bassem Ouni, and Muhammad Shafique. A survey of adversarial defenses in vision-based systems: Categorization, methods and challenges. *arXiv preprint arXiv:2503.00384*, 2025. 1
- [9] Hemang Chawla, Kishaan Jeeveswaran, Elahe Arani, and Bahram Zonooz. Image masking for robust self-supervised monocular depth estimation, 2023. 5
- [10] Zhaoyu Chen, Bo Li, Shuang Wu, Jianghe Xu, Shouhong Ding, and Wenqiang Zhang. Shape matters: deformable patch attack. In *European conference on computer vision*, pages 529–548. Springer, 2022. 5
- [11] Zitao Chen, Pritam Dash, and Karthik Pattabiraman. Jujutsu: A two-stage defense against adversarial patch attacks on deep neural networks. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, page 689–703, New York, NY, USA, 2023. Association for Computing Machinery. 6, 8
- [12] 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)*, pages 886–893 vol. 1, 2005. 1, 5
- [13] 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. 3, 5, 8
- [14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231, 1996. 7
- [15] D Freedman, R Pisani, and R Purves. *Statistics*. 2007. ISBN: 0-393970-833, 1978. 2, 3
- [16] Shivam Garg, Nandish Chattopadhyay, and Anupam Chattopadhyay. Robust perception for autonomous vehicles using dimensionality reduction. In *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1516–1521. IEEE, 2022. 3
- [17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 5
- [18] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015. 1
- [19] Amira Guesmi, Ruitian Ding, Muhammad Abdullah Hanif, Ihsen Alouani, and Muhammad Shafique. Dap: A dynamic adversarial patch for evading person detectors, 2023. 7
- [20] Amira Guesmi, Muhammad Abdullah Hanif, Bassem Ouni, and Muhammad Shafique. Physical adversarial attacks for camera-based smart systems: Current trends, categorization, applications, research challenges, and future outlook. *IEEE Access*, 2023. 1
- [21] Amira Guesmi, Muhammad Abdullah Hanif, Ihsen Alouani, Bassem Ouni, and Muhammad Shafique. Ssap: A shape-sensitive adversarial patch for comprehensive disruption of monocular depth estimation in autonomous navigation applications. *arXiv preprint arXiv:2403.11515*, 2024. 1, 5, 6
- [22] Sahand Hariri, Matias Carrasco Kind, and Robert J Brunner. Extended isolation forest. *IEEE transactions on knowledge and data engineering*, 33(4):1479–1489, 2019. 2
- [23] Chaoxiang He, Xiaojing Ma, Bin B Zhu, Yimiao Zeng, Hanqing Hu, Xiaofan Bai, Hai Jin, and Dongmei Zhang. Dorpatch: Distributed and occlusion-robust adversarial patch to evade certifiable defenses. In *NDSS*, 2024. 5
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 5
- [25] Yu-Chih-Tuan Hu, Jun-Cheng Chen, Bo-Han Kung, Kai-Lung Hua, and Daniel Stanley Tan. Naturalistic physical adversarial patch for object detectors. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7828–7837, 2021. 1, 5, 7, 8
- [26] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6084–6092, 2019. 1
- [27] Dan Karmon. Lavan: Localized and visible adversarial noise. In *International Conference on Machine Learning*, 2018. 5, 8

- [28] Alexander Levine and Soheil Feizi. (de) randomized smoothing for certifiable defense against patch attacks. *Advances in Neural Information Processing Systems*, 33:6465–6475, 2020. [5](#), [6](#), [8](#)
- [29] Fei-Fei Li, Marco Andreeto, Marc’Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. [5](#)
- [30] Xiang Li and Shihao Ji. Generative dynamic patch attack. *arXiv preprint arXiv:2111.04266*, 2021. [5](#)
- [31] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008. [2](#)
- [32] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868. IEEE, 2019. [1](#)
- [33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. [5](#)
- [34] 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. IEEE, 2019. [5](#), [6](#), [8](#)
- [35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [5](#)
- [36] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Towards hiding adversarial examples from network interpretation. *CoRR*, abs/1812.02843, 2018. [7](#)
- [37] Bilel Tarchoun, Anouar Ben Khalifa, Mohamed Ali Mahjoub, Nael Abu-Ghazaleh, and Ihssen Alouani. Jedi: Entropy-based localization and removal of adversarial patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4087–4095, 2023. [6](#), [8](#)
- [38] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. *CoRR*, abs/1904.08653, 2019. [1](#), [5](#), [6](#)
- [39] Xiaoling Xia, Cui Xu, and Bing Nan. Inception-v3 for flower classification. In *2017 2nd international conference on image, vision and computing (ICIVC)*, pages 783–787. IEEE, 2017. [5](#)
- [40] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwal, and Prateek Mittal. {PatchGuard}: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2237–2254, 2021. [5](#), [6](#), [8](#)
- [41] 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)*, pages 441–444. IEEE, 2006. [5](#)
- [42] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021. [5](#)