Grado en Ingeniería Matemática e Inteligencia Artificial

Trabajo Fin de Grado

Estudio de Robustez de Clasificadores frente a Ataques Dirigidos

One-Pixel Attack

Autor: Francisco Javier Ríos Montes

Director: Emanuel Gastón Mompó Pavesi

Madrid, Junio de 2025

"Where a model breaks is where it speaks the loudest about how it understands the world"

Universidad Pontificia Comillas
Escuela Técnica Superior de Ingeniería (ICAI)
Grado en Ingeniería Matemática e Inteligencia Artificial

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título **Estudio de Robustez de Clasificadores frente a Ataques Dirigidos mediante *One-Pixel Attack*** en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2024/25 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: **Francisco Javier Ríos Montes** Fecha: 11/06/2025

Autorizada la entrega del proyecto

El Director del Proyecto

Fdo.: **Emanuel Gastón Mompó Pavesi** Fecha: 11/06/2025

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Emanuel, for his patience and guidance throughout these months, as well as my parents for their continuous support. This project is dedicated to my late grandfather, Agustin, for his advice and counsel during these 4 years, without which I wouldn't have been able to complete this thesis.

Universidad Pontificia Comillas
Escuela Técnica Superior de Ingeniería (ICAI)
Grado en Ingeniería Matemática e Inteligencia Artificial

Robustness of Classifiers against Targeted Attacks with One-Pixel Attack

Author: Francisco Javier Ríos Montes
Supervisor: Emanuel Gastón Mompó Pavesi
Collaborating Entity: Universidad Pontificia Comillas - ICAI

Abstract

This thesis replicates and extends the One-Pixel Attack (OPA), an adversarial strategy that perturbs a single pixel to mislead Convolutional Neural Netowkrs (CNNs), as introduced by (Su, Vargas, and Sakurai 2019). We begin by reproducing the original results and addressing specific methodological inconsistencies in the generation of OPA samples. Then, the analysis is extended in two directions: First, we investigate whether a model's classification accuracy correlates with its vulnerability to such attacks; second, we assess whether modifying the theoretical receptive field of early convolutional layers affects the influence of the perturbed pixel. All experiments are conducted on models based on CNNs and trained on the CIFAR-10 dataset. Code for reproducing the experiments as well as pretrained model weights can be found at GitHub.

Keywords: CNN, Adversarial Attacks, OPA, Black-box, Robustness-accuracy Trade-off

1. Introduction

Over the last decade, Convolutional Neural Networks (CNN)-based image classifiers have achieved remarkable success across diverse fields such as medical imaging, autonomous driving, and industrial quality control. However, these models are known to be vulnerable to adversarial attacks, where small input perturbations can lead to incorrect predictions. These can be classified into white-box or black-box, depending on the knowledge about the model the attacker uses.

Our work focuses on a particularly striking case of the latter kind: the One-Pixel Attack (*OPA*, as per its acronym), where modifying a single pixel in an image can lead the network to misclassify it with high confidence. This counterintuitive result raises important questions regarding the *decision processes* of CNNs: What features do these models rely on in order to ’understand’ an image? What is the geometry of their decision boundaries? May they be excessively complex, thus indicating overfitting? In conclusion, to what extent can we trust their predictions in safety-critical settings?

2. Project Overview

We use the **CIFAR-10** dataset, consisting of 60,000 32×32 RGB images across 10 classes, which are already labeled. Our first goal is to replicate the results reported in the original OPA paper by training CNN classifiers to match the stated accuracies and applying the Differential Evolution (DE) algorithm to generate adversarial samples. After reproducing the baseline results, we explore two extended settings: First, we test whether model accuracy correlates with robustness to OPA, a topic with mixed findings in the literature (Tsipras et al. 2019).

Then, we investigate whether changing the theoretical receptive field (TRF) in early layers - by increasing kernel sizes and adjusting paddings - affects the pixel's impact. While some evidence suggests broader receptive fields may dampen localized perturbations (Suresh, Nayak, and Kalyani 2024), thus attenuating their propagation, others argue that theoretical changes do not always translate into effective influence propagation (Luo et al. 2017)

3. Methodology

3.1. Classification Models

Throughout the paper, we will deal with three main models: AllConv, Network-in-Network (NiN) and VGG16. All of them are based on the CNN architecture with certain caveats: NiN and AllConv both avoid the use of dense layers for the final classification (thus, relying on less number of parameters), replacing them with 1×1 convolutions. VGG16, on the other hand, is significantly deeper and heavier, due to the use of three large fully connected layers.

In the table below, the main properties of each model are described. For a deeper explanation of the individual structures, please refer to the Appendix (A).

Model	Accuracy (%)	# Parameters
AllConv	85.6	1,369,738
NiN	87.2	2,736,458
VGG16	83.3	33,638,218

Table 1: Comparison of the models by Original Accuracy (Su, Vargas, and Sakurai 2019) and Complexity

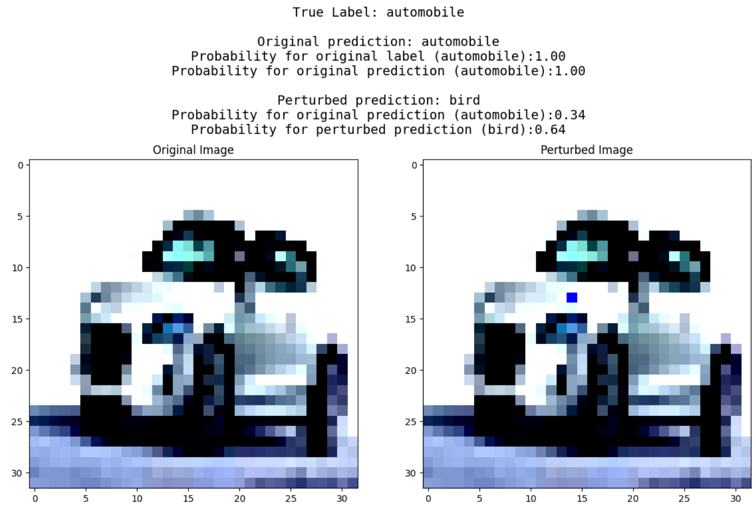
3.2. Adversarial Attack Algorithm

One-Pixel Attack employs Differential Evolution (DE) to generate adversarial samples by modifying a single pixel in the image. This black-box method requires only the output probabilities from the target model, without access to its internal parameters or gradients (thus, making it a more realistic setting).

In the OPA framework, each candidate solution represents a potential pixel modification, defined by its position and RGB values, thus having 5 individual parameters. DE then iteratively evolves these candidates to minimize the model's confidence in the correct class (untargeted attack) or maximize the assigned probability to a given class (targeted attack).

4. Results

With respect to the replication of the original results, our algorithm performs successfully 4.44%, 6.56% and 7.48% of the targeted attacks on AllConv, NiN and VGG16, respectively. In terms of the untargeted success rates, we find 28%, 38.67% and 41% of natural images can be perturbed as to lead the models to misclassify them. In the image below, we visualize one example of the targeted kind: By modifying one pixel of an image originally labeled as automobile, specifically the one at position $[14, 17]$, we are able to *mislead* AllConv so that it outputs bird as its prediction.



```
True Label: automobile

Original prediction: automobile
Probability for original label (automobile):1.00
Probability for original prediction (automobile):1.00

Perturbed prediction: bird
Probability for original prediction (automobile):0.34
Probability for perturbed prediction (bird):0.64
```

Figure 1: Example of a CIFAR-10 Image Before vs. After Successful OPA (AllConv)

Regarding our two additional scenarios, our findings can be summarized in the following way:

1. The variants we develop with higher accuracy are less vulnerable to both targeted and untargeted attacks than their original counterparts. Specifically, improvements of 4.36 and 6.36 percentage points in test accuracy (for AllConv and VGG16) yield a relative reduction in vulnerability to targeted attacks of 8 and 15 percentage points, respectively. Thus, accuracy and robustness to attacks are not strictly in opposition. Instead, other factors such as model capacity must be taken into account.

2. Increasing the TRF in early layers by increasing kernel sizes (but maintaning spatial dimensions constant by means of higher padding) results in diminished adversarial robustness. Particularly, increasing it to $k = 5$ implies a change in targeted attack success from 3.06% to 3.78%. Taking this even further only diminishes the performance more drastically, with the variant with $k = 7$ incurring in a targeted success rate of 5%

5. Conclusions

Starting from the results obtained by Su, Vargas, and Sakurai (2019), we confirm that minimal pixel-level perturbations can mislead even deep models such as VGG16 and NiN. With our extended studies, we demonstrate that while there may be a certain tension between test accuracy and robustness to adversarial attacks, the specific relationship must be studied carefully, since its variables seem to be more complex than just accuracy. Our hypothesis points at overfitting being the main actor. Moreover, our results on modifications intended to expand early-layer receptive fields (i.e., increasing TRF via larger kernels) underscore the vital distinction between TRF and the effective receptive field, which only reinforces the importance of focusing on truly impactful representations rather than purely architectural expansions.

References

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

2. Tsipras, Dimitris et al. *Robustness May Be at Odds with Accuracy.* arXiv: 1805.12152 url: https://arxiv.org/abs/1805.12152 May 2018

3. Suresh, Janani, Nancy Nayak, and Sheetal Kalyani. *First line of defense: A robust first layer mitigates adversarial attacks.* arXiv: 2408 . 11680 [cs.LG]. url: https://arxiv.org/abs/2408.11680. August 2024

4. Luo, Wenjie et al. (2017). *Understanding the Effective Receptive Field in Deep Convolutional Neural Networks.* arXiv: 1701.04128 [cs.CV]. url: https://arxiv.org/abs/1701.04128 January 2017

Contents

1 Introduction

1.1 Motivation and Relevance of Adversarial Robustness

In recent years, deep learning models have achieved remarkable success in a wide range of tasks, from medical diagnosis and autonomous driving to surveillance and biometric security. However, their vulnerability to adversarial attacks - often imperceptible perturbations that can completely alter the model's output - presents a critical threat to the reliability and trustworthiness of these systems. The One-Pixel Attack (OPA), in particular, demonstrates just how fragile models can be: a single manipulated pixel can be enough to mislead a classifier. This fragility is not just a technical flaw; it hinders the deployment of AI systems in high-stakes environments, where these apparently innocuous errors can carry significant economic, legal, or even life-threatening consequences.

This thesis contributes to the broader effort of understanding and mitigating such vulnerabilities by analyzing and extending the OPA on multiple convolutional neural network architectures. We go beyond replication, studying how architectural choices and training strategies - such as model accuracy or the theoretical receptive field - affect adversarial robustness. Our work offers practical insights for building more secure AI systems; ones which our society can truly trust. From a societal standpoint, ensuring that machine learning models behave reliably and as they're expected to is foundational to increasing public trust and ensuring the ethical deployment of AI in sensitive domains. From a research standpoint, it emphasizes the growing need to evaluate models not only in terms of accuracy but also robustness and interpretability.

1.2 Goals

The primary goal of this paper is to replicate and extend the One-Pixel Attack (OPA) on deep convolutional classifiers trained on CIFAR-10, in order to better understand the factors that influence adversarial robustness. Specifically, we aim to evaluate how variations in model accuracy and architectural design — such as increasing the theoretical receptive field (TRF) — impact a model's susceptibility to targeted attacks. Through quantitative analysis and class-wise breakdowns, we seek to identify structural and training-related patterns that affect the decision boundaries of neural networks, ultimately contributing to the design of more robust and interpretable models.

1.3 Planning and Economic Feasibility

This project was carried out over a period of several months, following a structured research-development-evaluation cycle, which is compressed in the Gantt Diagram shown below.

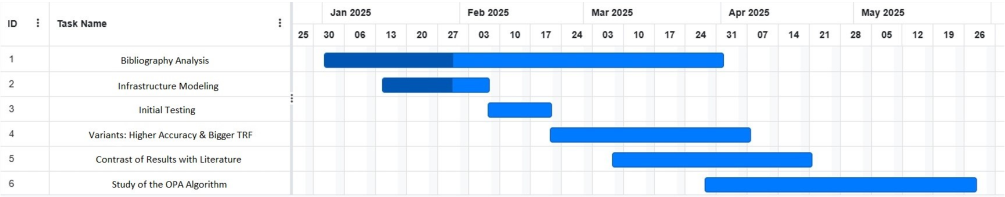


Figure 2: Gantt Diagram of our Thesis Development

From an economic perspective, the resources required for this work were minimal thanks to the use of open-source tools and publicly available datasets. All experiments, as well as the training of the models, were conducted using Google Colab Pro environments with a single NVIDIA T4 GPU (16GB VRAM), leveraging the libraries PyTorch and NumPy for the attack generation and model training software. Moreover, the project offers cost-effective insights into model vulnerability without requiring access to gradients or internal model parameters. The methodological framework is adaptable to different datasets or model architectures with minimal reconfiguration, thus offering high return on effort for real-world deployment evaluations.

1.4 Structure of the Paper

Our paper is organized as follows. In Section 2, we review the related works, including prior studies on the OPA, as well as literature addressing the relationships between model accuracy, receptive field design, and robustness. Section 3 outlines the technical methodology, detailing the dataset, model architectures, and the adversarial sample generation process. Section 4 focuses on the replication of the original OPA results, highlighting methodological inconsistencies encountered during reproduction. Section 5 presents the results of our extended experiments, which analyze the role of accuracy and receptive field size in adversarial robustness. In Section 6, we briefly discuss the broader implications of our findings. Section 7 presents potential directions for future research. Finally, Section 8 concludes the paper and summarizes our contributions.

2 Related Works

2.1 One-Pixel Attacks and Black-Box Methods

With the introduction of the OPA, Su, Vargas, and Sakurai (2019) demonstrated surprising vulnerabilities in State-of-the-Art Deep Neural Networks (DNNs) under extreme sparsity conditions. Subsequent works have improved and broadened the method's scope: Some focused on enhancing the optimization process behind the attack (Zhou, Agrawal, and Manocha 2022; Nam et al. 2024), while others evaluated cross-domain applicability (Nguyen et al. 2021) and generalization across architectures (Ghosh et al. 2021).

Beyond pixel-level heuristics, other approaches to black-box adversarial attacks have emerged. Decision-based attacks - such as Brendel, Rauber, and Bethge (2018) - craft adversarial inputs using only the model's final label, reflecting even more limited, albeit realistic, scenarios where confidence scores are unavailable. Together, these works define a spectrum of black-box techniques - ranging from pixel-level evolution to gradient-free boundary and score-based search - each reflecting distinct assumptions about attacker capabilities and model access.

2.2 Differential Evolution in Adversarial Contexts

Differential Evolution is a stochastic, population-based optimization strategy introduced by Storn and Price (1997), and it has been effectively adapted to the adversarial image attacks under black-box constraints. Beyond the implementation by Su, Vargas, and Sakurai (2019), several works harness the gradient-free approach to perform adversarial attacks. Z. Lin et al. (2023) allow for dynamic adjustments of the algorithm parameters and operation strategies. Their method significantly improved attack success rates on CIFAR-10 and MNIST - surpassing OPA in both efficiency and perturbation sparsity.

2.3 Accuracy-Robustness Trade-off

A growing body of research has revealed that, contrary to what intuition might lead us to, high accuracy on a given dataset does not guarantee robustness to adversarial perturbations. Moreover, recent studies identify a trade-off between the two, which implies the two must be weighed together in order to reach an optimal balance. In this respect, Tsipras et al. (2019) argue that this trade-off "is a consequence of robust classifiers learning fundamentally different feature representations than standard classifiers". These features, additionally "tend to align better with salient data characteristics and human perception".

Thus, a model's high accuracy may provoke a certain lack of robustness to adversarial attacks. At this point, though, we must introduce a caveat. Some authors argue that the main actor behind this reduction of robustness is the predominant use of over-parametrized networks and training-until-convergence attitudes (Rice, Wong, and Kolter 2020). These common practices "surprisingly do not unduly harm the generalization performance of the classifier", but "overfitting to the training set does in fact harm robust performance to a very large degree in adversarially robust training". Thus, the trade-off identified by previous authors may be avoided by using techniques such as early stopping.

2.4 Theoretical vs. Effective Receptive Field and Robustness

Understanding how architectural choices influence both the spatial context captured by Convolutional Neural Networks (CNNs) and their vulnerability to adversarial perturbations is crucial for building robust models. These architectures are often analyzed in terms of their TRF, which grows with kernel size and network depth (the greater the kernel size used in the convolution operation, the broader 'perspective' our network should have on the image). However, Luo et al. (2017) introduced the concept of the effective receptive field (ERF) and demonstrated that, in practice, only a central Gaussian-shaped core of the TRF strongly influences neuron activations. Additionally, the ERF grows sublinearly and is significantly smaller, meaning most pixels within the TRF have negligible impact.

Recent works by Ding et al. (2022) revisited the use of large early-layer kernels. They found that increasing kernel sizes not only expands the ERF but also shifts features biases from texture towards shape. Shape-biased features have been previously linked with improved adversarial robustness (Geirhos et al. 2022).

These findings tend to the following conclusions: while expanding TRF (e.f., via larger kernels or deeper stacks) increases theoretical context, actual influence is constrained by the ERF's Gaussian core - limiting robustness benefits. Therefore, simply increasing the TRF doesn't automatically translate to higher robustness to adversarial attacks.

3 Methodology

3.1 Models and Datasets

3.1.1 CIFAR-10

The original One-Pixel Attack paper applied its method to several datasets: **CIFAR-10**, **CIFAR-100** and **ImageNet**. Due to limitations in both computational resources and time, our study focuses exclusively on the first. Composed of 60,000 32×32 RGB images across 10 classes [1], *CIFAR-10* is a benchmark dataset of labeled images, commonly used for supervised learning tasks. It offers a well-established benchmark for evaluating adversarial robustness. Importantly, its low resolution - each image consisting of only 1,024 pixels - makes it particularly suitable for this work. Given that the OPA is constrained to modifying a single pixel, higher-resolution datasets would significantly reduce the likelihood of successful perturbations, especially in a black-box setting.

3.1.2 Overview of NiN, AllConv and VGG16 Architectures

As per the models, we focus on CNN-based architectures. Even though some exciting new architectures have come up in the last years, such as Vision Transformers (Dosovitskiy et al.

[1] List of classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck

2021), CNNs remain crucial in all image-related tasks. Therefore, studying the robustness of these models is key in order to understand the algorithms we use on a daily basis.

The Network-in-Network (NiN) architecture (M. Lin, Chen, and Yan 2014), innovated the traditional CNN by incorporating 1 × 1 convolutions - acting like per-pixel multilayer perceptrons - immediately after larger 5 × 5 kernels. This replacement of fully connected layers drastically reduces parameter count and improves spatial generalization.

The AllConv model (Springenberg et al. 2015) simplifies, in a similar way, common CNN architectures by removing pooling options entirely and replacing them with strided convolutions. Thus, the authors argue, replacing pooling by convolutions "can also be seen as learning the pooling operation rather than fixing it", which allows models to determine optimal ways to reduce spatial dimensions while preserving important information. Added to this, it enables a more easier inversion of the process, since max-pooling is an inherently non-invertible operation, so approximations are needed.

Last, the VGG16 model (Simonyan and Zisserman 2015) is the most classic DNN out of the three. Composed of 13 convolutional layers (using 3 × 3 kernels) and 5 Max-pooling layers, it is a really deep architecture that delivered strong performance on ImageNet, and it scales effectively to smaller datasets like CIFAR-10. To better understand the computational complexity and baseline performance of each architecture, Table 2 summarizes the original authors' reported accuracies and parameter counts on CIFAR-10.

Model	**Accuracy (%)**	**# Parameters**
AllConv	85.6	1,369,738
NiN	87.2	2,736,458
VGG16	83.3	33,638,218

Table 2: Model Performance Comparison on CIFAR-10

3.2 Adversarial Attack Setup

This section details our black-box adversarial attack method against CNNs on CIFAR-10. We follow the protocol established by Su et al. (2017), adapting it in key respects to ensure methodological rigor.

3.2.1 One-Pixel Attack Definition

The One-Pixel Attack is an extreme case of adversarial perturbation: only a **single pixel** may be modified arbitrarily. Thus, it represents a constrained L_0 optimization, albeit it doesn't restrict the magnitude of this modification. Given an input image $x \in \mathbb{R}^{32\times32\times3}$ a classifier f (in our case, a neural network), let "true" represent the original class. the attacker seeks a perturbation $\delta(x)$ such that $\|\delta(x)\|_0 = 1$ and $f(x + \delta(x)) \neq f(x)$. For targeted mode, additionally, let "target" be the desired label. Then, we can define both scenarios as optimization problems:

1. **Targeted Attack**

$$\begin{aligned} \max_{\delta(x)^*} \quad & f_{\text{target}}(x + \delta(x)) \\ \text{s.t.} \quad & \|\delta(x)\|_0 \leq 1 \end{aligned}$$

2. **Untargeted Attack**

$$\begin{aligned} \min_{\delta(x)^*} \quad & f_{\text{true}}(x + \delta(x)) \\ \text{s.t.} \quad & \|\delta(x)\|_0 \leq 1 \end{aligned}$$

with f_{target} and f_{true} being the probability assigned by the model to the original label and the target label, respectively, and $\delta(x)^*$ being the optimal perturbation.

3.2.2 Differential Evolution Algorithm

Finding the optimal pixel to modify is a highly non-differentiable search problem due to the discrete and combinatorial nature of selecting a pixel and altering its RGB values. This, combined with the black-box setting where gradients are inaccessible, makes traditional gradient-based methods unsuitable. As a means of tackling this issue, OPA employs **Differential Evolution** (Storn and Price 1997), a population-based, gradient-free optimizer. DE maintains a *population* of candidate perturbations, where each candidate encodes a pixel location and its new RGB values: $[x, y, r, g, b]$ [2]. In the mutation step, child candidates are created by perturbing three parents:

$$x_i(g+1) = x_{r_1}(g) + F(x_{r_2}(g) - x_{r_3}(g)), \quad r_1 \neq r_2 \neq r_3 \tag{3.1}$$

where $F \in [0, 2]$ is the mutation scale (a parameter set to 0.5 in the OPA paper), which controls the magnitude of the differential variation $x_{r_2} - x_{r_3}$. The indices r_1, r_2, r_3 represent randomly selected candidates of the population, thus $r_i \sim U(1, 400) \quad \forall i \in \{1, 2, 3\}$, with 400 being the population size. This specific variant is known as **rand1bin**: 'rand' indicates that the base vector is selected randomly from the population, '1' means a single difference vector is used for mutation, and 'bin' refers to the binary crossover method [3].

After mutation, **greedy selection** is performed, where a new candidate replaces its parent if it achieves higher attack fitness. Depending on whether the attack is targeted or untargeted,

[2] In order to optimize the search, positions and RGB values are normalized to the range $[0, 1]$ with perturbations then being cast to the correct range ($[0, 31]$ for position and $[0, 255]$ for pixel values).

[3] It must be noted, though, that (Su, Vargas, and Sakurai 2019) omit the Crossover operation, in order to simplify the algorithm. Therefore, we also omit the procedure, following their method.

this fitness function differs: In the former case it takes the form of the probability assigned by the model to the target label while, in the latter, it's the negative (applying objective negation) of the probability assigned to the original label.

Although we initially planned to use a custom implementation of Differential Evolution, we ultimately opted for Python's **Torchattacks** (TA) library (Kim 2020), which provides a dedicated class for the One-Pixel Attack. This decision was driven primarily by computational efficiency: Our custom implementation (Ríos 2025), required approximately 30-40 seconds to generate a single adversarial example, whereas the TA-based implementation consistently produced results in roughly 15 seconds per sample. This trade-off allowed us to process larger batches of images under strict time and computational constraints without compromising the attack's core methodology.

Reproducibility Note: Due to the nature of the attack implementation and its reliance on a stochastic, population-based optimized without explicit seed control, strict reproducibility of exact perturbations is not feasible. Nevertheless, the algorithm's overall behavior is consistent across runs.

3.2.3 Constraints and Evaluation Metrics

To ensure fair comparisons, we preserved key constraints from the original study:

- **Pixel limit:** Only one pixel may be modified ($\|\delta_0\| = 1$), with RGB values matching the *uint8* format of CIFAR-10 ($[0, 255]$).
- **Population**: Initial population size is set to 400 and, after each iteration, 400 new candidates are chosen.
- **Early stopping criteria:** The maximum number of iterations is set to 100. This process can be terminated early if success metrics are met (e.g., the output for the best current candidate matches our objective).

We evaluate success using the following metrics, consistent with Su et al:

1. **Targeted Attack Success Rate:** Ratio of successful perturbations to total attempts. A successful perturbation, in this context, is thus defined as a modified natural image which is predicted as *target label* by the model.
2. **Untargeted Attack Success Rate:** Instead of evaluating untargeted attacks on a separate setting, we extrapolate the previous to these results. This is, if a natural image can be perturbed to, at least, one other class (different, of course, to the original one), its untargeted attack is successful.
3. **Original Label Success Rate**: For each model, we include the breakdown of successes depending on the original label.
4. **Target Label Success Rate**: For each model, we also obtain the proportion of fruitful attacks depending on the target label.

3.3 Experiment Pipeline and Implementation

With the aim of clarifying what the results we provide in the following sections imply, we will dedicate the following paragraphs to explaining the basis for our experiments. These can be divided between the different pipelines used in the two scenarios we studied: Replication of the original results and analysis of accuracy and TRF with respect to their impact on models' robustness to adversarial attacks

3.3.1 Replication of Original Results

To construct a suitable test set, we select 300 **correctly classified images**[4] per model from the CIFAR-10 Test Dataset. Even though the original authors used a sample size of 500, we restricted this due to computational limitations. Additionally, we enforce class balance (thus, choosing 30 different images out of each class) in order to avoid bias towards certain classes.

3.3.2 Analysis of Accuracy and Theoretical Receptive Field

In this case, we can further divide the methods used in the two cases:

- **Accuracy analysis:** In order to assess whether there's any correlation between accuracy and robustness to attacks (i.e., whether more accurate models are more/less vulnerable to them), we train two of the three models (AllConv and VGG16) until convergence, obtaining an improvement in accuracy of 4.36% and 6.36%, respectively. While not enormous in absolute terms, we believe it's a significant improvement in order to assess any difference.

- **TRF analysis:** Starting from the original AllConv architecture, we define two structural variants by modifying the kernel size in the first convolutional block (the one in direct contact with the image fed as input). In specific, we replace the original size ($k = 3$) with $k = 5$ and $k = 7$, thus increasing the theoretical receptive field of early layers. This, as was already discussed previously, could theoretically either propagate the signal further, diminish the perturbed pixel's *weight* or, if the change in TRF doesn't translate to a change in ERF, not have any significant impact

In both cases, we use a subset of the images used for the base models' attacks. This is, we restrict the original set of 300 images to those that are, additionally, correctly classified by the variants. At the same time, we maintain the class balance enforcement. The result is two subsets of 200 correctly classified, class-balanced images. The first is used to evaluate the high-accuracy version of **VGG16**, while the second serves both the high-accuracy and TRF-modified versions of **AllConv**. In the graphic below, we provide a visualization of this process for **AllConv**.

[4] In section 4.2, we explain why this is crucial

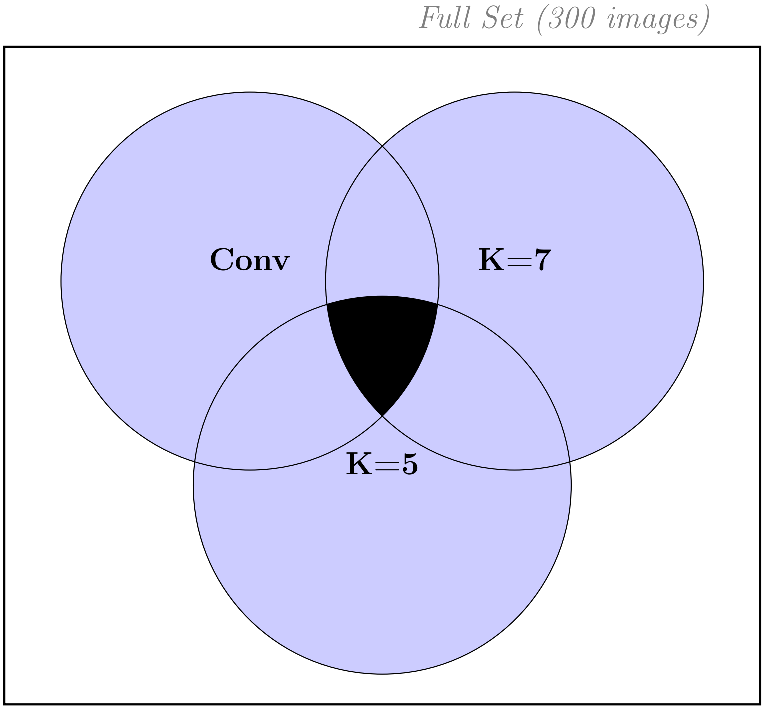


Figure 3: Venn diagram showing correctly classified image overlap across 3 model variants. The rectangle area corresponds to the original set of 300 images. Conv corresponds the higher-accuracy version of AllConv, and $K = 5$, $K = 7$ the TRF Variants. The darkest region (overlap between all sets) is the subset of 200 images used for all comparisons.

Preliminary Limitations: While our methodology aims to replicate and extend the OPA, several practical constraints apply. We omit visualization of decision boundaries due to time constraints and defer broader visualizations (for example, transferability or cross-architecture generalization) to future work. These considerations are discussed further in Section 7.

4 Critical Comparison with Su, Vargas, and Sakurai (2019)

At this point, we must highlight several differences between the methodology stated by the original authors and the one we use, as well as certain decisions that may inadvertently bias the results.

4.1 Dataset Differences

The former authors' attack was performed on two versions of CIFAR-10: the original one (that which we described previously) and Kaggle's dataset. The latter retains the general class structure and image resolution, but contains 300,000 images which have been *augmented*, this is, modified in several ways (duplication, rotation, clipping, blurring...). Therefore, they argue, the former dataset implies an scenario that is "more limited since the images contain much less practical noise" [5]. Therefore, the target CNNs can have higher classification and confidence which definitely makes the attack harder" (Su, Vargas, and Sakurai 2019). In the following paragraphs, we will discuss what this implies and its importance, as well as why it's our opinion that this may not actually be the cause for the difference in performance.

4.2 Model Accuracy and Filtering Criteria

In order to compare both scenarios in similar conditions, the authors trained the three models explained previously (AllConv, NiN and VGG16) under both datasets, independently, to have similar accuracies. Then, Differential Evolution was applied on all of them, in order to obtain the adversarial samples. The only difference in methodology between the two settings was that, in the one using Kaggle's dataset, they allowed misclassified images to be included as inputs to the attack, while they didn't in the original. In the table below, we include a copy of their results.

	AllConv	**NiN**	**VGG16**
Targeted	3.41%	4.78%	5.63%
Untargeted	22.67%	32.00%	30.33%
Confidence	54.58%	55.18%	51.19%

	AllConv	**NiN**	**VGG16**
Targeted	19.82%	23.15%	16.48%
Untargeted	68.71%	71.66%	63.53%
Confidence	79.40%	75.02%	67.67%

Figure 4: Comparison of (Left) Original and (Right) Kaggle CIFAR-10 OPA Results (Su, Vargas, and Sakurai 2019)

4.3 Implications for Attack Difficulty

Figure 4 illustrates the stark difference in One-Pixel Attack success rates between models evaluated on the two datasets. The authors attributed this gap to "higher classification confidence" caused by "less practical noise" in the original images (Su, Vargas, and Sakurai 2019), even though the models were trained to identical accuracies across both datasets. However this explanation fails to account for a far more simple explanation: the inclusion of already misclassified samples in the generation of attacks.

Specifically, since the original paper did not filter for correctly classified inputs when attacking the Kaggle-based models, the a attack may succeed more easily simply because some inputs were already on the wrong side of decision boundaries. The literature corroborates that robustness assessments may not be dependable when evaluations include such misclassified cases, which are inherently closer to adversarial decision thresholds. For example, (Wang et al. 2019) showed that misclassified examples significantly inflate adversarial success during evaluation, and that distinguishing between correct and incorrect base classifications is critical for accurate robustness measurement.

[5] This "practical noise" refers to perturbations that might occur in realistic image capture, such as compression artifacts, noise or sensor defects

Thus, the increased attack effectiveness observed in the Kaggle experiments is more plausibly driven by dataset filtering, not noise characteristics. This insight reinforces our methodology of consistently evaluating attacks using only correctly classified samples, underpinning the validity of our subsequent robustness comparisons.

5 Experimental Results

5.1 Baseline Model Comparison

Given what we discussed in the previous section (4.3), in order to compare our results with the original paper's, we will only use the metrics provided with respect to the original version of the CIFAR-10 dataset. In the following paragraphs, we provide this comparison in which models' accuracies match those stated in Su et al.

5.1.1 Success Rates across NiN, AllConv and VGG16

The table below presents the success rates of adversarial attacks on the three baseline models. As stated previously, we conduct only targeted attacks explicitly. Metrics for untargeted attacks are obtained implicitly: if, for a given natural image, at least one of the nine targeted perturbations leads to a misclassification (i.e., a label different from the original), the untargeted attack is considered successful for that image. The evaluation is performed over the set of 300 images we presented in Section 3.3.1. Since each image is attacked to all nine possible target labels (excluding the true label), a total of 2,700 perturbation attempts are carried out per model.

Model	**Su et al. (2019)**		**Ours**		
	Targeted	**Untargeted**	**Targeted**	**Untargeted**	**# Successes**
AllConv	3.41 %	22.67 %	4.44 %	28.00 %	120
NiN	4.78 %	32.00 %	6.56 %	38.67 %	177
VGG16	5.63 %	30.33 %	7.48 %	41.00 %	202

Table 3: Comparison of OPA Results on the 3 Baseline Networks:
Original Paper vs. Our Implementation

From these results, we observe that our implementation achieves higher success rates across all models compared to the original paper, both in targeted and untargeted settings. The improvement ranges from approximately 1-2 percentage points in targeted attacks, and even more substantially (5-10 percentage points) in untargeted success.

Although the original authors didn't explicitly state the exact algorithm used to perform the adversarial attacks, several factors may explain the observed discrepancies: First, while our methodology replicates theirs closely, we rely on the *Torchattacks* library, which may include subtle optimizations in DE parameter handling. Second, the original paper does not mention enforcing class balance in the evaluation subset. If class imbalance were present, it could introduce bias toward more vulnerable or more robust classes, thus skewing the results. Finally, hardware and library updates could contribute to slightly different behaviors in floating-point computations or convergence dynamics - especially considering the six-year gap between implementations.

Despite these potential sources of divergence, the relative differences between models (e.g., VGG16 consistently being the most vulnerable) are preserved, reinforcing the robustness of the attack strategy itself.

5.1.2 Success Rate Breakdown by Original/Target Label

Below we summarize the most and least vulnerable classes for each model, considering both the original-label and target-label perspectives. These extremes help us infer qualitative properties of the decision boundaries. The full per-class success-rate tables are provided in Appendix B.

Model	Role	Most Vulnerable	(%)	Most Robust	(%)
AllConv	Original	deer	10.00	automobile	1.11
	Target	dog	12.22	horse	1.85
NiN	Original	cat	13.70	horse	2.22
	Target	airplane	16.67	horse	2.59
VGG16	Original	deer	18.89	dog	2.22
	Target	dog	20.37	deer	1.85

Table 4: Most and Least Vulnerable Classes per Model
(as Original and Target Labels)

These class-wise breakdowns reveal several consistent patterns. First, the class *dog* stands out as the most common successful target across models (in the case of NiN, it is the second). This may suggest that all models attribute certain features to "dogs" that may be shared by a broad range of classes. Prior work shows that some classes define disproportionately large or smoothly curved surfaces in feature space, making them easier destinations for perturbed inputs (He, Li, and Song 2018)

In contrast, the class *horse* is robust, generally, both as a target and as an original label, possibly due to distinctive visual features or well-separated representation boundaries. Interestingly, *deer* shows high vulnerability as an original class but low vulnerability as a target, particularly in VGG16. This counter-intuitive asymmetry suggests that while the model has difficulty defending its original classification of "deer", it rarely confuses other classes **into** it.

These results can be interpreted in several ways, of which we would like to highlight two:

1. **Geometric Interpretation:** In terms of decision boundary geometry, a class that is highly vulnerable *as a target* - e.g., *dog* - likely occupies a broad or easily accesible

region of the output space, so many perturbed inputs are close to its classification boundary. On the other hand, a class that is highly robust as a target - e.g., *horse* - may lie deep inside its own decision region, far from the boundary separating it from others. Consequently, perturbations must cross large distances in feature space to be mapped to it, making such transitions less likely under deeply constrained attacks (e.g., OPA).

2. **Relation to Type I/II Errors**: In terms of hypothesis testing, this symmetry aligns with certain concepts. Misclassifying a sample from another class *into* a given target class is analogous to a Type I error (false positive) while failing to correctly classify an input of a class into its true label is akin to a Type II error (false negative). In our scenario, classes like *deer* tend to suffer from frequent Type II errors - i.e., failing to retain the original prediction under small perturbations - while rarely being a Type I error (few samples are misclassified into deer). This reflects a common bias in classifier behaviors: in practice, training loss functions - e.g., cross-entropy - often penalize Type II errors more heavily, emphasizing correct classification of true labels, but may allow regions of the output space to be more permissive as attractors for perturbed inputs.

These insights suggest that some classes act as *gravitational wells* in prediction space, pulling in perturbed inputs, while others are inherently better defended or isolated - a property that may not be easily detected by accuracy metrics alone.

5.2 Accuracy vs. Robustness Analysis

Let's briefly revisit, first, the scenario under which we intend to study the accuracy vs. robustness trade-off. In order to keep all other variables constant, we use two subsets of 200 images, one for the AllConv variant and one for VGG16's.[6]

5.2.1 Setup: Low-Accuracy vs. high-Accuracy Variants

The original authors don't mention their criteria for choosing the accuracy objectives. Therefore, we don't have any original assumptions as to the level of over-fitting the models may inadvertently fall upon. Thus, we let the models train to convergence, obtaining the following improvements:

1. **AllConv**: We improve the accuracy from 85.6% to **89.96**%, thus obtaining an absolute improvement of 4.36 percentage points

2. **VGG16**: We improve the accuracy from 83.3% to **89.66**%, thus obtaining an absolute improvement of 6.36 percentage points.

[6] Therefore, in the coming results, the success rates for the baseline models are restricted to the respective subset of 200 images

5.2.2 Statistical Results and Interpretation

Model	Variant	Targeted (%)	Untargeted (%)	# Successes
AllConv	baseline (85.6 % acc.)	3.06	22.00	55
	high-acc. (89.96 % acc.)	2.83	20.00	51
VGG16	baseline (83.3 % acc.)	7.33	39.50	132
	high-acc. (89.66 % acc.)	6.22	33.50	112

Table 5: OPA success rates on low and high-accuracy variants

Both AllConv and VGG16 exhibit reduced success rates for targeted and untargeted OPA after additionaly training. While the absolute differences appear modest, the relative improvement in targeted attack success is notable: AllConv demonstrates an 8% decrease (3.06 → 2.83) while VGG16 shows a 15% reduction (7.33 → 6.22). These strengthenings are particularly significant given the baseline challenge of targeted OPA, where even marginal improvements can require substantial perturbation optimization.

5.2.3 Revisiting the Trade-Off Debate

As introduced in Section 2.3, there is a certain tension between accuracy and adversarial robustness. Increasing standard accuracy often comes at the cost of reduced adversarial robustness (Tsipras et al. 2019). However, this relationship comes with a caveat, which our findings show: Both AllConv and VGG16 improved in clean accuracy while exhibiting modest - but consistent - decreases in OPA success rates. The divergence from the theoretical trade-off proposed by Tsipras et al. could be due to several key differences in assumptions:

- **Attack model limitations**: The One Pixel Attack is a highly constrained L_0-bounded method, and may not fully capture the robustness metrics used in L_∞-norm analyses typically assumed in theoretical studies.
- **Dataset size and subset stability**: Our experiments were performed on balanced subsets of only 200 images per model variant. While carefully controlled, the results may not generalize across the full distribution.
- **Accuracy vs. Overfitting**: While both models increased their test accuracy, there is no immediate indication of overfitting. This distinction is important: overfitting often leads to memorization rather than generalization, which in turn reduces robustness. Our improvements, being consistent across test subsets, suggest genuine gains in generalization rather than mere fitting of the training data.

In light of the above, we interpret the results as indicating that **accuracy and robustness are not strictly in opposition**, especially when models are still operating below their capacity threshold and not yet overfitting, as indicated by Rice, Wong, and Kolter (2020)

5.3 Receptive Field and Attack Propagation

5.3.1 Model Variants

In order to compare the effects of increasing the TRF on attack robustness, we develop variants for the **AllConv** model. Specifically, we increase the first convolutional block's kernel sizes, increasing padding to ensure spatial structure is kept constant [7].

5.3.2 Results and Implications

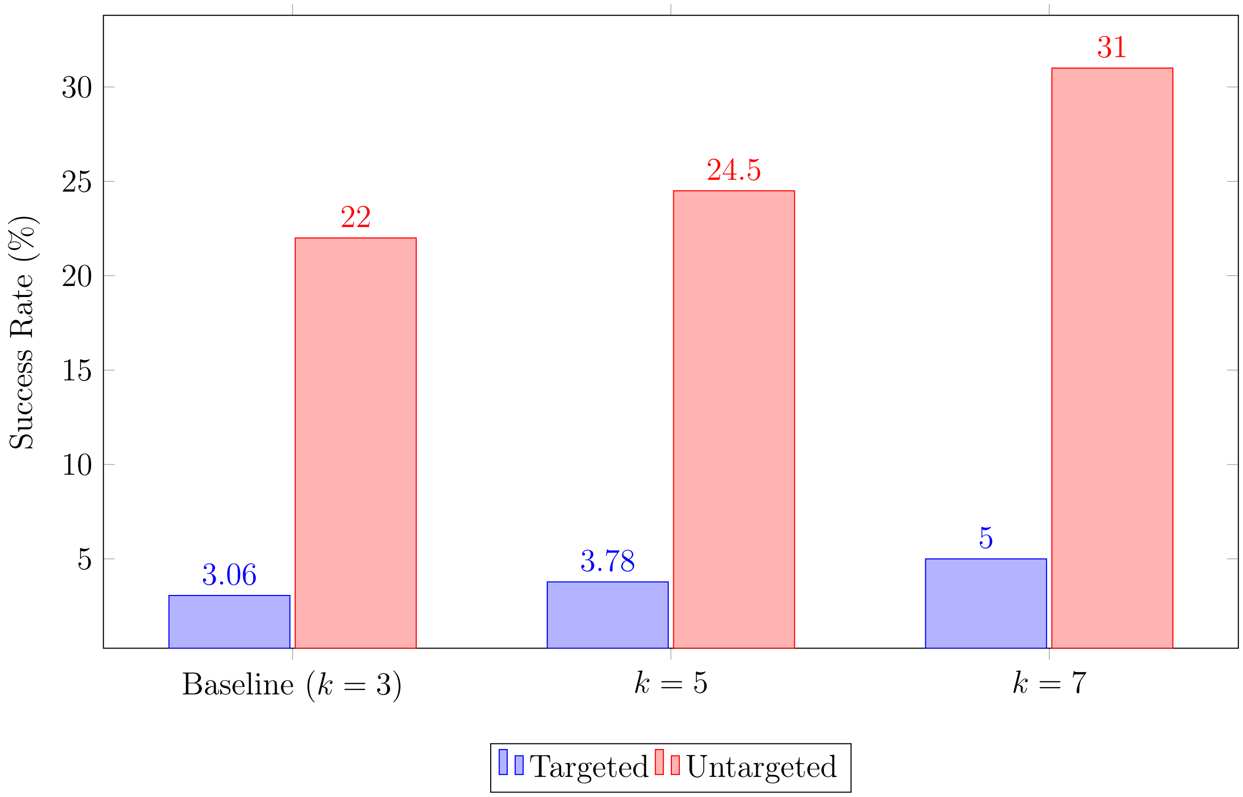


Figure 5: OPA success rates for AllConv with different kernel sizes

Our AllConv experiments show that increasing the kernel size from 3×3 (baseline) to 5×5 and 7×7 raises both targeted and untargeted OPA success rates, indicating diminished adversarial robustness at larger receptive fields.

5.3.3 Connection to ERF Literature

The results provided previously align with the evidence provided by Luo et al. (2017). Even though we increased the TRF, we can conclude these changes didn't translate to an increase

[7] The specific parameters and modified layers can be found in Appendix A

in ERF. The decline of our model's robustness may be explained by several factors, but our hypothesis is that increasing the kernel size encouraged the model to rely on less robust features. Thus, we weren't able to reach the required kernel sizes needed to harvest the improvements argued by Ding et al. (2022) [8]. Therefore, our results may not be directly transferrable to even higher kernel sizes.

6 Discussion

6.1 Summary of Findings

Our study replicates the original results of the One-Pixel Attack (OPA) on three convolutional architectures — AllConv, NiN, and VGG16 — confirming that all exhibit measurable vulnerability to single-pixel perturbations in a black-box setting. Beyond replication, we explore two key extensions: the relationship between classification accuracy and adversarial robustness, and the impact of increasing the TRF in early convolutional layers. In the first case, we observe that both AllConv and VGG16 maintain or even improve their robustness to OPA as test accuracy increases, suggesting that improved generalization does not necessarily come at the expense of adversarial resilience. In the second, we find that increasing kernel size (and thereby TRF) leads to higher, not lower, attack success rates—highlighting the crucial distinction between TRF and the effective receptive field (ERF), which governs the actual region influencing classification decisions.

These findings contribute to the broader understanding of robustness in deep neural networks by reinforcing the limitations of overly architectural interpretations of adversarial vulnerability. Our results support the idea that robustness improvements are not guaranteed by increasing model complexity or spatial field of view alone, and that specific design strategies — such as those targeting ERF expansion or feature-type shifts — may be more effective in practice. Additionally, by carefully isolating and controlling variables (e.g., test set filtering, kernel size adjustments, model accuracy), we provide a clearer empirical framework for future investigations into CNN interpretability and attack resistance.

6.2 Limitations of the Current Work

While the experimental design of this work was constructed to control for confounding factors, several limitations must be acknowledged. First, our experiments are constrained to a relatively small subset of the CIFAR-10 test set — only 300 images per model, and 200 for the variant comparisons. Although these subsets were balanced and filtered for correctness, the limited scope may restrict the generalizability of our conclusions across the full data distribution or to more complex datasets. Second, the attack considered is extremely specific: OPA restricts perturbations to a single pixel. While its simplicity provides valuable insight

[8] It must be noted that, in Ding et al., kernel sizes of up to 31 × 31 were used

into localized vulnerability, it may not fully capture the broader threat landscape posed by stronger or higher-dimensional attacks.

Another constraint arises from the architectural modifications explored. While we adjust kernel sizes to alter TRF in early layers, we do not experiment with more sophisticated mechanisms such as dilated convolutions or ERF visualization techniques. Consequently, we can only hypothesize—rather than directly measure—whether the ERF remained narrow, which would require gradient-based saliency maps or input attribution methods. Finally, although we strive for reproducibility, the use of stochastic optimizers like Differential Evolution (without seed control in Torchattacks) introduces slight randomness in outcomes, which may influence exact success rates across runs.

6.3 Revisiting the Trustworthiness of CNN Predictions

Our results call into question the often-assumed reliability of CNN predictions, particularly under adversarial settings. That a single pixel, chosen through a black-box optimization process, can cause consistent misclassification across multiple architectures is a striking reminder of the fragile boundary geometry learned by CNNs. The fact that deeper or more accurate models do not necessarily resolve this vulnerability — sometimes even being more vulnerable — further complicates the narrative that better performance equates to greater reliability. While our high-accuracy variants showed slight improvements in robustness, the effect was modest and may not generalize across all model types or perturbation settings.

This unpredictability has serious implications for the deployment of CNNs in safety-critical applications. If a model can be fooled through such minimal interventions, and if architectural choices like kernel size have inconsistent effects on vulnerability, it becomes imperative to reconsider the notion that high test accuracy alone is sufficient to guarantee trustworthy outputs. As our analysis suggests, and recent evidence backs (Singh et al. 2021), evaluation of a model quality must extend beyond predictive accuracy to include factors such as interpretability and adversarial robustness. The ERF framework provides a promising lens for developing architectures that prioritize meaningful feature reliance—potentially paving the way for models that are not only accurate, but also more reliably aligned with human-intuitive reasoning.

7 Future Work

7.1 Visualization and Interpretability

The constraints our current work relies on forbid us from experimenting, further, with interpretability methods. By integrating these visualization-based techniques - such as saliency maps, guided backpropagation or feature activation visualizations - the classifiers' decision processes could be understood with greater clarity. Thus, applying these tools may aid not only in debugging but also in improving adversarial robustness, highlighting unstable regions of feature space. Moreover, combining interpretability methods and adversarial analysis - an emerging area of interest in AI safety - may allow us to identify the most vulnerable components of our models, which would enable a more fine-grained adaptation of architectures.

7.2 Broader Metrics

In order to assess the results of our work, we have relied on rather straight-forward success-rate metrics - namely, the proportion of images successfully perturbed under targeted and untargeted attack settings. While these provide an immediate quantitative view of model vulnerability, they do not capture the full spectrum of robustness properties.

Future work should consider incorporating more expressive and informative metrics. FOr instance, robustness curves that track success rates as a function of allowed perturbation (e.g., restricting the magnitude of the perturbation or allowing for a higher number of pixels to be perturbed) could offer a more nuanced understanding of boundary geometry. Similarly, metrics such as adversarial confidence margins would allow us to quantify the strength of perturbations, beyond binary success/failure. These complimentary methods would enrich the evaluation of adversarial roubstness and offer more meaningful criteria for model selection in real-world deployments

8 Conclusions

This thesis validates and extends the findings of the One-Pixel Attack literature across multiple CNN architectures and improves the clarity on the methodology used. In replication, we confirm that minimal pixel-level perturbations can mislead even deep models such as VGG16 and NiN. In our extended studies, we demonstrate that while improving test accuracy can slightly improve robustness — suggesting better generalization — it does not fully eliminate adversarial vulnerability and does not necessarily correlate with broader adversarial resilience. Moreover, modifications intended to expand early-layer receptive fields (i.e., increasing TRF via larger kernels) led to **greater**, not less, vulnerability to OPA. This result underscores the vital distinction between **theoretical receptive fields** and **effective receptive fields**, reinforcing the importance of focusing on truly impactful feature representations rather than purely architectural expansions.

Our work highlights several practical insights:

- Boosting accuracy alone is insufficient as a surrogate for model reliability
- Receptive field modifications must be examined in terms of actual feature influence (ERF), not just theoretical reach
- Future evaluation frameworks should encompass interpretability-focused diagnostics and diverse performance metrics

Together, these findings contribute to the growing consensus that robust, trustworthy AI must be built and assessed on multi-faceted grounds — so that models that perform well under clean data also are resistant to adversarial pressure and exhibit understandable behavior.

References

Brendel, Wieland, Jonas Rauber, and Matthias Bethge (2018). *Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models*. arXiv: 1712.04248 [stat.ML]. URL: https://arxiv.org/abs/1712.04248.

Ding, Xiaohan et al. (2022). *Scaling Up Your Kernels to 31x31: Revisiting Large Kernel Design in CNNs*. arXiv: 2203.06717 [cs.CV]. URL: https://arxiv.org/abs/2203.06717.

Dosovitskiy, Alexey et al. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

Geirhos, Robert et al. (2022). *ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness*. arXiv: 1811.12231 [cs.CV]. URL: https://arxiv.org/abs/1811.12231.

Ghosh, Arka et al. (2021). *A Black-box Adversarial Attack Strategy with Adjustable Sparsity and Generalizability for Deep Image Classifiers*. arXiv: 2004.13002 [cs.CR]. URL: https://arxiv.org/abs/2004.13002.

He, Warren, Bo Li, and Dawn Song (2018). "Decision Boundary Analysis of Adversarial Examples". In: *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. Published on OpenReview. Vancouver, Canada. URL: https://openreview.net/forum?id=BkpiPMbA-.

Kim, Hoki (2020). "Torchattacks: A pytorch repository for adversarial attacks". In: *arXiv preprint arXiv:2010.01950*.

Lin, Min, Qiang Chen, and Shuicheng Yan (2014). *Network In Network*. arXiv: 1312.4400 [cs.NE]. URL: https://arxiv.org/abs/1312.4400.

Lin, Zhiyi et al. (2023). "Image Adversarial Example Generation Method Based on Adaptive Parameter Adjustable Differential Evolution". In: *Entropy* 25.3. ISSN: 1099-4300. DOI: 10.3390/e25030487. URL: https://www.mdpi.com/1099-4300/25/3/487.

Luo, Wenjie et al. (2017). *Understanding the Effective Receptive Field in Deep Convolutional Neural Networks*. arXiv: 1701.04128 [cs.CV]. URL: https://arxiv.org/abs/1701.04128.

Nam, Wonhong et al. (Apr. 2024). "RISOPA: Rapid Imperceptible Strong One-Pixel Attacks in Deep Neural Networks". In: *Mathematics* 12, p. 1083. DOI: 10.3390/math12071083.

Nguyen, Quoc et al. (July 2021). *OPA2D: One-Pixel Attack, Detection, and Defense in Deep Neural Networks*. DOI: 10.1109/IJCNN52387.2021.9534332.

Rice, Leslie, Eric Wong, and J. Zico Kolter (2020). *Overfitting in adversarially robust deep learning*. arXiv: 2002.11569 [cs.LG]. URL: https://arxiv.org/abs/2002.11569.

Ríos, Francisco Javier (2025). *Adversarial Attacks on Classifiers (OPA)*. GitHub Repository. URL: https://github.com/Javirios03/Adversarial_Attacks_Classifiers (visited on 06/11/2025).

Simonyan, Karen and Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs.CV]. URL: https://arxiv.org/abs/1409.1556.

Singh, Moninder et al. (2021). *An Empirical Study of Accuracy, Fairness, Explainability, Distributional Robustness, and Adversarial Robustness*. arXiv: 2109.14653 [cs.LG]. URL: https://arxiv.org/abs/2109.14653.

Springenberg, Jost Tobias et al. (2015). *Striving for Simplicity: The All Convolutional Net*. arXiv: 1412.6806 [cs.LG]. URL: https://arxiv.org/abs/1412.6806.

Storn, Rainer and Kenneth Price (Jan. 1997). "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". In: *Journal of Global Optimization* 11, pp. 341–359. DOI: 10.1023/A:1008202821328.

Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai (Oct. 2019). "One Pixel Attack for Fooling Deep Neural Networks". In: *IEEE Transactions on Evolutionary Computation* 23.5, pp. 828–841. ISSN: 1941-0026. DOI: 10.1109/tevc.2019.2890858. URL: http://dx.doi.org/10.1109/TEVC.2019.2890858.

Suresh, Janani, Nancy Nayak, and Sheetal Kalyani (2024). *First line of defense: A robust first layer mitigates adversarial attacks*. arXiv: 2408.11680 [cs.LG]. URL: https://arxiv.org/abs/2408.11680.

Tsipras, Dimitris et al. (2019). *Robustness May Be at Odds with Accuracy*. arXiv: 1805.12152 [stat.ML]. URL: https://arxiv.org/abs/1805.12152.

Wang, Yisen et al. (2019). "Improving Adversarial Robustness Requires Revisiting Misclassified Examples". In: *International Conference on Learning Representations (ICLR)*. OpenReview preprint. URL: https://openreview.net/forum?id=rklOg6EFwS.

Zhou, Tianxun, Shubhankar Agrawal, and Prateek Manocha (2022). *Optimizing One-pixel Black-box Adversarial Attacks*. arXiv: 2205.02116 [cs.CR]. URL: https://arxiv.org/abs/2205.02116.

A Models' Structure

This appendix details the specific layer-wise architecture of the three models used throughout our experiments: AllConv, NiN (Network in Network), and VGG16 (CIFAR-10 variant). Each table outlines the sequence of operations and parameter settings used. Output sizes are given in the notation $[W, H, C]$, with W and H being the height and width of the resulting feature maps and C the number of feature maps (number of filters applied in the layer).

AllConv

Layer	Output Size	Configuration
Conv1	32x32x96	3x3 conv (P=1), ReLU
Conv2	32x32x96	3x3 conv (P=1), ReLU
Conv3	32x32x96	3x3 conv (P=1), stride=2, ReLU
Dropout1	32x32x96	Dropout (p=0.5)
Conv4	16x16x192	3x3 conv, ReLU
Conv5	16x16x192	3x3 conv, ReLU
Conv6	16x16x192	3x3 conv, stride=2, ReLU
Dropout2	16x16x192	Dropout (p=0.5)
Conv7	8x8x192	3x3 conv, ReLU
Conv8	8x8x192	1x1 conv, ReLU
Conv9	8x8x10	1x1 conv, ReLU
GlobalAvgPool	1x1x10	Global average pooling
Softmax	10-dim	Classification layer

Table 6: AllConv model architecture. The three highlighted layers correspond to the ones we change for Section 5.3

With respect to the model variants we use to test our hypothesis on the *TRF vs. ERF* scenario, we modify the kernel size of the three first convolutional layers from $k = 3$ to $k = 5$ and $k = 7$. In order to maintain the spatial size constant throughout the network (in comparison with the baseline model), we take into account the following formula:

$$H_{out} = \left\lfloor \frac{H_{in} + 2P - K}{S} \right\rfloor + 1, \quad W_{out} = \left\lfloor \frac{W_{in} + 2P - K}{S} \right\rfloor + 1 \tag{A.1}$$

for $[W_{out}, H_{out}]$, $[W_{in}, H_{in}]$ the width and height of the output/input, and where K is the kernel size (assumed square for simplicity), S the stride, P the padding used and $\lfloor \cdot \rfloor$ the floor operation (round down to the nearest integer).

Therefore, if we increase K by 2 units, we must increase padding by 1 unit to keep the spatial dimensions of the layer's output constant. Thus, in the first variant - AllConv with $k = 5$ - the corresponding layers' padding is increased to $P = 2$ and in the second - AllConv with $k = 7$ - padding is increased to $P = 3$

Network in Network (NiN)

Layer	Output Size	Configuration
Conv1	32x32x192	5x5 conv, ReLU
MLPConv1	32x32x160	1x1 conv, ReLU
MLPConv2	32x32x96	1x1 conv, ReLU
MaxPool1	16x16x96	3x3 max pool, stride=2
Dropout1	16x16x96	Dropout (p=0.3)
Conv2	16x16x192	5x5 conv, ReLU
MLPConv3	16x16x192	1x1 conv, ReLU
MLPConv4	16x16x192	1x1 conv, ReLU
AvgPool2	8x8x192	3x3 avg pool, stride=2
Dropout2	8x8x192	Dropout (p=0.3)
Conv3	8x8x192	3x3 conv, ReLU
MLPConv5	8x8x192	1x1 conv, ReLU
MLPConv6	8x8x10	1x1 conv, ReLU
GlobalAvgPool	1x1x10	Global average pooling
Softmax	10-dim	Classification layer

Table 7: Network in Network (NiN) model architecture.

VGG16 (CIFAR-10 Variant)

Layer	Output Size	Configuration
Conv1_1	32x32x64	3x3 conv, ReLU
Conv1_2	32x32x64	3x3 conv, ReLU
MaxPool1	16x16x64	2x2 max pool
Conv2_1	16x16x128	3x3 conv, ReLU
Conv2_2	16x16x128	3x3 conv, ReLU
MaxPool2	8x8x128	2x2 max pool
Conv3_1	8x8x256	3x3 conv, ReLU
Conv3_2	8x8x256	3x3 conv, ReLU
Conv3_3	8x8x256	3x3 conv, ReLU
MaxPool3	4x4x256	2x2 max pool
Conv4_1	4x4x512	3x3 conv, ReLU
Conv4_2	4x4x512	3x3 conv, ReLU
Conv4_3	4x4x512	3x3 conv, ReLU
MaxPool4	2x2x512	2x2 max pool
Conv5_1	2x2x512	3x3 conv, ReLU
Conv5_2	2x2x512	3x3 conv, ReLU
Conv5_3	2x2x512	3x3 conv, ReLU
GlobalAvgPool	1x1x512	Global average pooling
FC	10-dim	Fully connected layer (10 classes)
Softmax	10-dim	Classification output

Table 8: VGG16 (CIFAR-10 variant) model architecture.

B Attack Sucess Rates by Original/Target Labels

B.1 Baseline Models

Class	AllConv		NiN		VGG16	
	Original	Target	Original	Target	Original	Target
Airplane	2.59	4.44	3.33	16.67	4.81	12.22
Automobile	1.11	4.81	3.33	3.70	4.07	3.70
Bird	5.93	4.44	8.15	10.00	7.41	12.59
Cat	9.63	4.07	13.70	7.04	14.81	8.15
Deer	10.00	2.59	8.52	4.07	18.89	1.85
Dog	2.59	12.22	6.67	10.37	2.22	20.37
Frog	2.96	3.70	5.19	2.96	7.78	4.07
Horse	3.33	1.85	2.22	2.59	6.67	2.22
Ship	2.96	3.70	11.48	2.96	5.19	4.44
Truck	3.33	2.59	2.96	5.19	2.96	5.19

Table 9: Class-wise Targeted Attack Success Rates (%) per Original and Target label

B.2 Accuracy Variants

Class	Original AllConv		Convergent AllConv	
	Original (%)	Target (%)	Original (%)	Target (%)
Airplane	2.59	4.44	2.22	3.89
Automobile	1.11	4.81	1.11	3.33
Bird	5.93	4.44	4.44	2.78
Cat	9.63	4.07	6.11	1.11
Deer	10.00	2.59	5.56	1.67
Dog	2.59	12.22	1.67	8.89
Frog	2.96	3.70	0.56	2.22
Horse	3.33	1.85	3.33	1.11
Ship	2.96	3.70	2.78	3.89
Truck	3.33	2.59	2.78	1.67

Table 10: Class-wise Targeted Attack Success Rates Across AllConv Variants (Original vs. High-Accuracy)

Class	**Original VGG16**		**Convergent VGG16**	
	Original (%)	Target (%)	Original (%)	Target (%)
Airplane	5.00	11.11	7.78	8.89
Automobile	5.00	3.33	3.33	2.22
Bird	7.78	14.44	9.44	9.44
Cat	15.56	6.67	13.33	8.33
Deer	19.44	1.11	9.44	2.22
Dog	2.22	21.67	2.78	16.67
Frog	7.78	4.44	6.11	5.56
Horse	5.56	2.22	2.22	1.67
Ship	3.89	3.89	6.11	2.22
Truck	1.11	4.44	1.67	5.00

Table 11: Class-wise Targeted Attack Success Rates Across VGG16 Variants (Original vs. High-Accuracy)

B.3 TRF Variants

Class	**AllConv (k=3)**		**AllConv (k=5)**		**AllConv (k=7)**	
	Original	Target	Original	Target	Original	Target
Airplane	2.22	3.89	0.56	5.56	7.22	2.78
Automobile	1.11	3.33	2.78	2.78	1.11	1.67
Bird	4.44	2.78	2.22	6.67	6.11	7.78
Cat	6.11	1.11	1.11	10.56	7.78	5.00
Deer	5.56	1.67	8.33	0.56	5.00	6.11
Dog	1.67	8.89	7.78	4.44	9.44	11.67
Frog	0.56	2.22	2.22	2.22	1.67	3.89
Horse	3.33	1.11	5.00	0.00	6.11	3.89
Ship	2.78	3.89	6.11	1.67	4.44	1.67
Truck	2.78	1.67	1.67	3.33	1.11	5.56

Table 12: Class-wise Targeted Attack Success Rates (%) for AllConv Variants with Kernel Sizes $k = 3, 5, 7$