MSID: A Multi-Scale Diffusion-Based Inpainting Defense Against Adversarial Attacks

IN5000: Final Project

Andrei Popovici

TUDelft

MSID: A Multi-Scale Diffusion-Based Inpainting Defense Against Adversarial Attacks

by

Andrei Popovici

to obtain the degree of Master of Science

Master of Science in Computer Science
Track: Data Science & Technology
Specialization: Cyber Security

at the Delft University of Technology,
Faculty of Electrical Engineering, Mathematics and Computer Science,
to be publicly defended on Thursday, 6 February 2025, at 11:00 AM.

Student number:	5853060	
Thesis committee:	Prof. Dr. G. Smaragdakis,	Thesis advisor, TU Delft
	Dr. K. Liang,	Daily supervisor, TU Delft
	Dr. J. Decouchant,	TU Delft
Project Duration:	December 2023 - December 2024	

TU Delft

Preface

This thesis represents the culmination of a journey, one filled with challenges, discoveries, and the unwavering support of mentors, colleagues, and friends. As I stand at the threshold of this new chapter, I am filled with a profound sense of gratitude for the individuals who have shaped not only this work, but also my growth as a researcher.

I am deeply grateful to my committee members for their invaluable contributions, guidance, and insightful advice throughout this thesis project. Their expertise and critical feedback have been instrumental in refining my research and shaping the final form of this document. I would especially like to express my sincere appreciation to Dr. Kaitai Liang and Prof. Dr. Georgios Smaragdakis for their assistance with the process of paper submission. Dr. Liang's meticulous attention to detail and insightful suggestions significantly improved the clarity and precision of my thesis. I am also grateful to Dr. Jérémie Decouchant for taking the time to attend my defense and offer his support.

From the very first day of this journey, Yanqi Qiao has been a constant source of encouragement and knowledge. I am incredibly thankful for his willingness to address my questions, no matter how large or small, and for his firm guidance throughout this process. His support has been truly invaluable. I would also like to extend my gratitude to Dazhuang Liu for his valuable feedback on my thesis paper, which greatly strengthened the final product.

The path of research is rarely linear, it is a winding road paved with both moments of joy and periods of intense focus. The support I have received has been the compass that guided me through the uncertainties and the fuel that propelled me forward. This thesis is not just a culmination of my individual efforts, but a testament to the power of collaboration, mentorship, and the shared pursuit of knowledge. It is with sincere appreciation that I offer this work, hoping it reflects the invaluable contributions of those who have so generously shared their time and expertise.

Andrei Popovici
Delft, January 2025

Summary

This thesis paper addresses the vulnerability of Deep Neural Networks (DNNs) to adversarial attacks. We introduce Multi-Scale Inpainting Defense (MSID), a novel adversarial purification method leveraging a pre-trained diffusion denoising probabilistic model (DDPM) for targeted perturbation removal. MSID employs a four-step process: (1) multi-scale superpixel segmentation, (2) occlusion sensitivity map generation at multiple scales to identify important regions for inference, (3) targeted inpainting using the DDPM, and (4) artifact removal using Variance Preservation Sampling. We investigate the effectiveness of diffusion-based inpainting for robust defense, the impact of multi-scale occlusion sensitivity mapping, and the robustness of MSID against a set of adversarial attacks, including color-based attacks. Our experiments demonstrate that MSID outperforms existing adversarial purification methods, achieving robustness improvements of up to 5.42% on CIFAR-10 and 10.75% on ImageNet against AutoAttack, with further gains against PGD and unseen attacks, while maintaining high standard accuracy. This paper, to the best of our knowledge, is the first to apply DDPM inpainting for targeted adversarial purification and demonstrates its effectiveness in purifying a range of adversarial attacks.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AP	Adversarial Purification
AT	Adversarial Training
BPDA	Backward Pass Differentiable Approximation
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DCT	Discrete Cosine Transform
DDIM	Denoising Diffusion Implicit Model
DDPM	Denoising Diffusion Probabilistic Model
DNN	Deep Neural Network
EBM	Energy-Based Model
EOT	Expectation Over Transformation
FGSM	Fast Gradient Sign Method
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HQS	Half-Quadratic Splitting
MCMC	Markov Chain Monte Carlo
ODE	Ordinary Differential Equation
OOD	Out-of-Distribution
PGD	Projected Gradient Descent
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue channels
SDE	Stochastic Differential Equation
SLIC	Simple Linear Iterative Clustering
SVM	Support Vector Machine
VAE	Variational Autoencoder
ViT	Vision Transformer
VPS	Variance Preservation Sampling
WRN	WideResNet
XAI	Explainable AI

Symbols

Symbol	Definition
x	Input image
X	Input images set
d	Dimensionality of the input space
f_θ	DNN with parameters θ
K	Number of classes
δ	Adversarial perturbation
x'	Adversarial example
ϵ	Perturbation budget (maximum allowed perturbation magnitude)

Symbol	Definition
g_γ	Purification module with parameters γ
T	Number of timesteps
β_t	Variance schedule at timestep t
α	$1 - \beta_t$
$\bar{\alpha}$	Product of α_s from $s = 1$ to t
μ_γ	Learned mean of the reverse process parameterized by γ
$\sum_\gamma$	Learned covariance of the reverse process parameterized by γ
q	Imputer
M	Binary mask
λ	Conditioning guidance strength
u	Resampling iterations
ζ	Noise level
τ	Inverse strength
γ	Step size
L	Refinement Iterations (per timestep)

1 Introduction

Deep neural networks (DNNs) have achieved remarkable success in image classification, yet their vulnerability to adversarial attacks remains a significant concern [52, 68, 98, 3]. These attacks, often imperceptible perturbations to input images, can cause misclassifications, posing serious threats to safety-critical applications [36, 99]. Adversarial attacks can be broadly categorized into white-box attacks, where the attacker has full knowledge of the target model's architecture and parameters, gray-box attacks, where the attacker has partial knowledge, such as access to the model's training data or architecture but not the specific parameters or defense used and black-box attacks, where the attacker has limited or no such knowledge. Within these categories, attacks can be further classified based on their perturbation constraints (e.g., l_0, l_2, l_∞ norms) and goals (e.g., targeted misclassification, untargeted misclassification).Examples of attacks include the fast, white-box Fast Gradient Sign Method (FGSM) [36] and the Projected Gradient Descent (PGD) attack [65], a more powerful iterative extension of FGSM. This requires the development of robust defense mechanisms. Figure 1.1 demonstrates the effectiveness of MSID, our purification method, against two prominent adversarial attacks, cAdv [9] and PGD.

Existing defenses fall primarily into two categories: adversarial training (AT) and adversarial purification (AP). AT [36, 126, 65] improves robustness by training DNNs using adversarial examples. However, it often suffers from overfitting to known attacks [78], struggles with unseen attacks [53] and can degrade standard accuracy while increasing computational cost [101, 25]. AP, on the other hand, pre-processes input images to remove adversarial perturbations before inference [95, 90, 122]. Often leveraging generative models [86, 88, 34, 92, 75, 42, 123, 110], AP offers a plug-and-play advantage, requiring no retraining of the classifier and potentially generalizing to unseen attacks. However, AP typically exhibits lower standard accuracy compared to AT [102, 16]. Furthermore, achieving a balance between preserving image semantics and effectively removing perturbations remains a challenge, especially for large-scale datasets [110].

Diffusion Models (DMs) are a class of generative models that learn to synthesize data by reversing a diffusion process, gradually transforming a data sample into pure noise and then learning to reverse this process to generate new samples. Recently, DMs have shown remarkable performance in image generation, often outperforming even Generative Adversarial Networks (GANs) in terms of sample quality [43, 94]. Their inherent denoising process naturally aligns with the goal of purification, and their stochasticity offers potential for robust stochastic defenses [55]. However, existing diffusion-based AP methods, such as those in [75, 61, 110], are vulnerable to color-based attacks due to the sensitivity of DDPMs to chromatic manipulations [75]. This, combined with the limitations of AT in handling unseen attacks and maintaining standard accuracy [96, 53, 101], reveals a significant gap: the need for a defense that simultaneously maintains high standard accuracy, effectively removes adversarial perturbations, and provides robustness against a diverse range of attacks, including color-based and unseen attacks.

This thesis addresses this gap by introducing Multi-Scale Inpainting Defense (MSID), a novel adversarial purification method. MSID performs targeted perturbation removal through a four-step process: (1) Multi-scale superpixel segmentation to capture both coarse and fine-grained image features; (2) Occlusion sensitivity map generation at multiple scales to identify perturbation-sensitive regions; (3) Targeted inpainting using a pre-trained DDPM to restore these regions; and (4) Artifact removal using Variance Preservation Sampling (VPS) to ensure a natural final image. We hypothesize that MSID will improve resistance to color-based attacks and achieve state-of-the-art robust accuracy.

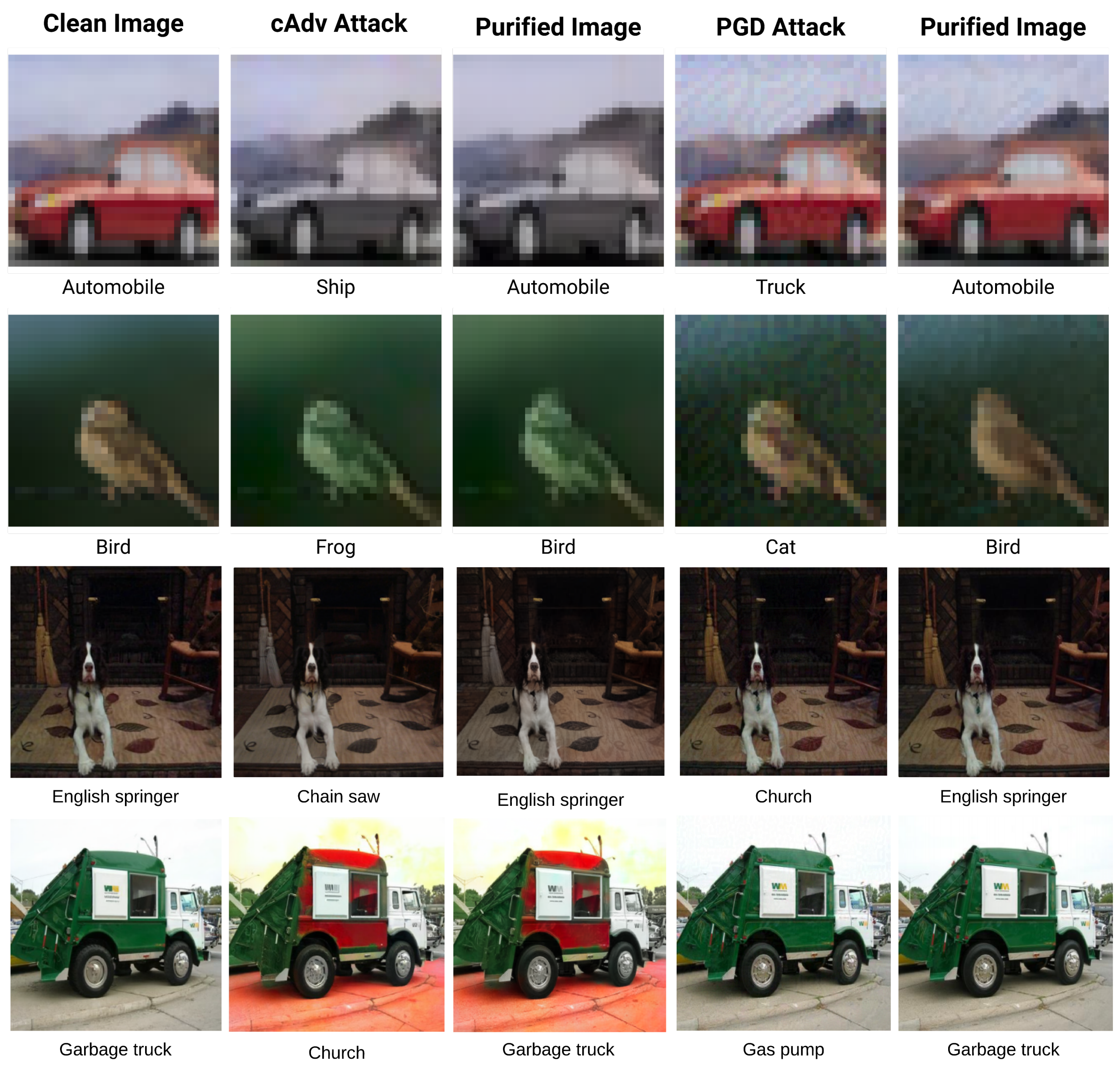


Figure 1.1: Illustration of MSID's purification effectiveness. Each row presents an original image (leftmost), followed by adversarial examples crafted using two attack methods (cAdv and PGD) and the corresponding purified outputs from MSID. MSID proves promising purification capabilities, effectively mitigating the adversarial perturbations introduced by the attacks

1.1. Research questions

This thesis investigates the vulnerability of DNNs to adversarial attacks and explores novel defense mechanisms based on diffusion-based inpainting and explainable AI (XAI) techniques. The research is guided by the following research questions:

RQ1: How can diffusion-based inpainting be leveraged to develop a more robust defense against adversarial attacks on image classifiers?

This question motivates the development of the MSID, exploring the potential of diffusion models to restore semantically meaningful content while removing adversarial perturbations. The focus is on utilising the generative capabilities of diffusion models to purify images before inference.

RQ2: Can a multi-scale approach to occlusion sensitivity mapping improve the identification and removal of adversarial perturbations?

This question examines the core contribution of MSID, which employs multi-scale superpixel segmentation and generates occlusion sensitivity maps at different levels of detail. The goal is to determine whether this multi-scale analysis improves the accurate localization of adversarial perturbations compared to single-scale version of the method.

RQ3: What methods can be used to generate a binary mask from the sensitivity map that accurately isolates regions that require inpainting for effective adversarial purification in MSID?

This question investigates the process of transforming the continuous-valued sensitivity map into a binary mask capable of guiding the inpainting process. The objective is to identify the most relevant regions for inpainting, thereby isolating the areas most likely to contain the adversarial perturbations. This research question explores and compares two distinct approaches for achieving this segmentation: thresholding and clustering. Thresholding techniques analyse individual pixel sensitivity values, labeling pixels exceeding a predetermined threshold as potentially perturbed, indicating a high contribution to the erroneous classification. Conversely, clustering methods consider the sensitivity map holistically, grouping pixels based on shared characteristics and spatial proximity to outline regions of potential perturbation.

RQ4: How does the choice of imputation strategy during occlusion sensitivity map generation influence the robustness of MSID against adversarial attacks?

This research question investigates the generation of the occlusion sensitivity map, which aims to identify the regions within the image that contribute most significantly to the classification. Specifically, it examines the impact of different imputation methods, used to fill the occluded regions during this sensitivity analysis on the overall robustness of the defense. The occlusion process involves temporarily removing portions of the image, and the method used to fill these occlusions can significantly influence the resulting sensitivity map and the subsequent identification of these critical regions. This research question explores three specific imputation strategies: (1) Zero-value imputation, filling the occluded regions with black pixels; (2) Histogram-based imputation, which occludes superpixels with a constant value sampled from the color histogram of the image; and (3) Blurring-based imputation, which replaces the occluded superpixels with a blurred version of the surrounding image content. These strategies represent three distinct approaches to filling the missing information, ranging from simple constant value filling (zero-value) to more context-aware methods (histogram and blurring). Different imputation strategies can introduce artifacts or biases that may affect the accuracy of the generated sensitivity map. This question will help us evaluate the effects of these three imputation strategies on the performance of MSID.

RQ5: Does MSID provide better robustness against a wider range of adversarial attacks, including unseen attacks and color-based attacks, compared to existing state-of-the-art defenses?

This question evaluates the effectiveness of MSID by benchmarking its performance against both established and novel attack strategies. The comparison encompasses a range of attack types, including white-box and black-box attacks, strong adaptive attacks and color-based attacks, to assess the generalization capabilities and overall robustness of the proposed defense. The aim is to demonstrate that MSID offers superior or comparable performance to state-of-the-art.

1.2. Contributions

The main contributions of this work are as follows:

- This paper, to the best of our knowledge, is the first to apply DDPM inpainting for adversarial purification, enabling targeted removal of perturbations while preserving benign image features, unlike classical diffusion models which operate globally.
- We propose a novel defense technique which combines occlusion sensitivity maps to identify potentially perturbed regions in adversarial images. This allows precise identification and removal of adversarial perturbations, improving the robsutness compared to methods that lack targeted restoration.
- MSID shows promising adversarial robustness against color-based attacks, a challenging type of attacks. On CIFAR-10, MSID achieves 75.19% robust accuracy and on ImageNet, 64.84%.
- Extensive experiments show that our method outperforms previous AP methods. On CIFAR-10, MSID improves robust accuracy by up to 5.42% against AutoAttack and 2.49% against PGD, while maintaining a competitive 90.86% standard accuracy. On ImageNet, the improvements reach 10.75% against AutoAttack, alongside a 76.39% standard accuracy. Furthermore, MSID shows significantly better robustness against unseen attacks, with gains of up to 36.9%.

1.3. Thesis organization

This thesis is structured as follows:

Chapter 2 (Preliminaries) provides the necessary background information on deep neural networks, adversarial attacks, adversarial training, adversarial purification and diffusion models. This chapter establishes the foundational concepts upon which the research is built. Chapter 3 (Literature Review) examines the existing literature on adversarial defense mechanisms. It explores the increasing threat posed by these attacks, highlights the importance of robust defenses, and discusses various defense strategies, including adversarial training and purification methods. This chapter identifies a research gap that motivates the proposed approach. Chapter 4 (Methodology) details the proposed defense mechanism. It explains the motivation behind the approach and provides a comprehensive description of the framework, including the generation of occlusion sensitivity maps, pixel grouping, feature occlusion, multi-scale superpixel segmentation, map fusion, clustering, DDPM-based inpainting, incorporation of contextual cues and artifact removal. Chapter 5 (Experimental Setup) describes the experimental setup used to evaluate the proposed defense. This includes details on the datasets, the defense and classifier architectures, evaluation metrics and the adversarial attacks used. Implementation details, including hyperparameters and code availability, are also provided. Chapter 6 (Results and Discussion) presents the experimental results and discusses their implications. This chapter compares the proposed defense to state-of-the-art methods, evaluates its performance against various attacks. This chapter also answers the research questions and provides an ablation study to analyse the contribution of the defense's different components. Chapter 7 (Limitations and Future Work) discusses the limitations of the proposed approach and outlines potential directions for future research. Chapter 8 (Conclusion) summarizes the key findings of the thesis and reiterates the contributions of the work.

2 Preliminaries

2.1. Deep neural networks for image classification

Deep Neural Networks (DNNs) have revolutionised image classification, achieving outstanding performance across various benchmarks [51]. DNNs are complex, multi-layered structures designed to learn patterns from data. At their core are artificial neurons, interconnected units that process and transmit information. These neurons are organised into layers, with each layer performing a specific transformation on the data. During training, DNNs learn by adjusting the weights of these connections to minimize a loss function, which measures the difference between the network's predictions and the ground truth labels. This process, known as backpropagation, iteratively adjusts the weights based on the gradients of the loss function. This section provides a brief overview of DNNs, focusing on Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), two architectures used in this thesis.

2.1.1. Convolutional Neural Networks (CNNs)

CNNs are designed to exploit the spatial structure of images. They use convolutional layers, which learn local patterns through filters that slide across the input image. These filters, characterized by their size, stride, and padding, effectively detect features like edges, corners, and textures. The output of a convolutional layer is a feature map, which is then often downsampled by pooling layers (e.g., max pooling or average pooling) to reduce computational complexity and increase robustness to small spatial shifts. Pooling operations aggregate information within local regions, further contributing to spatial invariance. After multiple convolutional and pooling layers, the extracted features are flattened and passed through fully connected layers, which combine these features and perform the final classification. Activation functions, such as the Rectified Linear Unit (ReLU), introduce non-linearity between layers. This non-linearity is important for enabling the network to learn complex, non-linear relationships in the data.

Popular CNN architectures, such as ResNet [41], WideResNet [124], and MobileNetV2 [87], which are used in this thesis, leverage these components with varying configurations and optimizations to achieve high accuracy and efficiency. ResNet, for instance, introduces skip connections to address the vanishing gradient problem, a common issue in deep networks, enabling the training of very deep architectures. WideResNet expands upon ResNet by increasing the width (number of channels) of the convolutional layers, which provides a richer feature representation. MobileNetV2 employs depthwise separable convolutions, a computationally efficient alternative to standard convolutions, reducing the number of parameters and computations, making it suitable for mobile and embedded applications. A typical CNN architecture is illustrated in Figure 2.1.

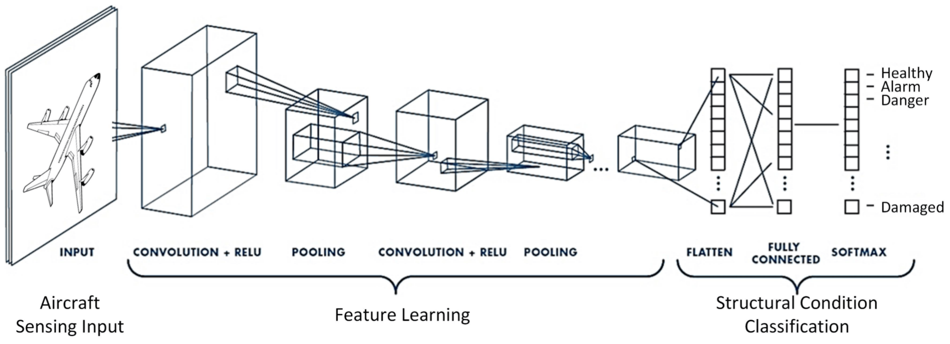


Figure 2.1: Architecture of a typical CNN for image classification. The input image is processed through a series of convolutional and pooling layers, extracting hierarchical features. Convolutional layers apply learnable filters to detect patterns, while pooling layers downsample the feature maps. ReLU activation functions introduce non-linearity. The flattened feature representation is then passed through fully connected layers for final classification using a softmax function. Reprinted from Tabian et al. [100]

2.1.2. Vision Transformers (ViTs)

Vision Transformers (ViTs) [26] offer an alternative approach to image classification, adapting the transformer architecture [105], originally designed for natural language processing, to the visual domain. ViTs process images by dividing them into a sequence of non-overlapping patches. These patches are then flattened and linearly projected into embedding vectors, similar to word embeddings in natural language processing. Positional encodings are added to these embeddings to retain spatial information, which is essential because the transformer architecture itself is permutation-invariant. These embedded patches are then fed into a transformer encoder, as depicted in Figure 2.2.

The core component of the transformer encoder is the self-attention mechanism. Self-attention allows the network to weigh the importance of different image patches in relation to each other, effectively capturing long-range dependencies and global context. Multi-head attention enhances this by performing self-attention multiple times in parallel with different learned linear projections (heads), allowing the model to attend to different aspects of the input simultaneously. Each attention head produces a set of attention weights, which are then aggregated and used to combine the input patch embeddings. Following the self-attention mechanism, the transformer encoder typically includes a feed-forward network, applying non-linear transformations to further process the information.

ViTs have shown competitive performance on image classification tasks, particularly when trained on large datasets [26]. They excel at capturing global context and long-range dependencies, offering a different inductive bias compared to CNNs. In this thesis, we also evaluate the performance of MSID with a ViT architecture on ImageNet, demonstrating its applicability to transformer-based models.

This overview provides the necessary background on DNNs for image classification, focusing on the architectures relevant to this thesis. The subsequent chapters will explore the specifics of how these architectures are used and evaluated within the context of adversarial defense. In particular, the vulnerability of DNNs to adversarial attacks develops from their complex decision boundaries and reliance on subtle features, making them susceptible to carefully crafted perturbations.

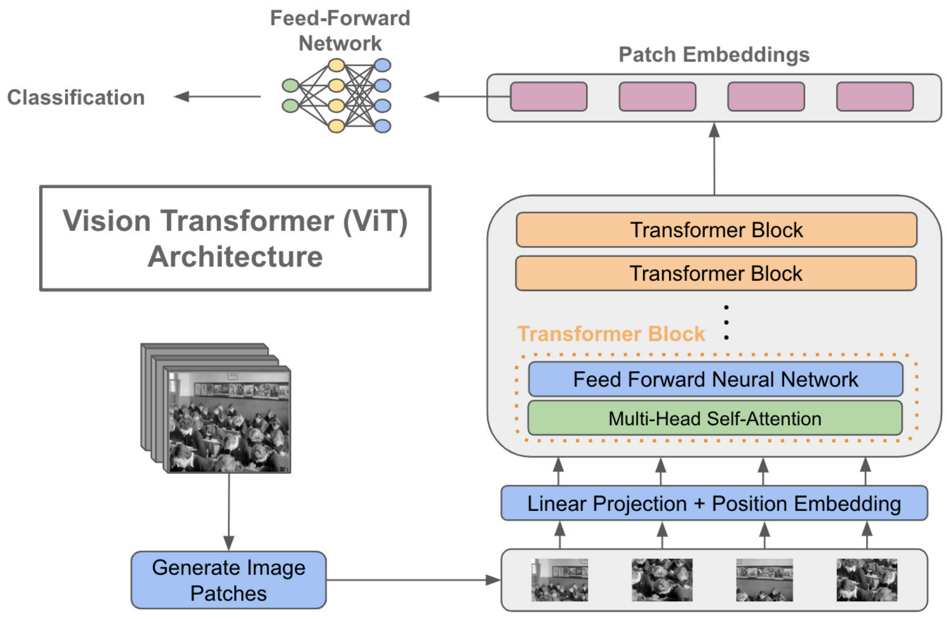

Feed-Forward Network
Classification
Patch Embeddings
Vision Transformer (ViT) Architecture
Transformer Block
Transformer Block
Transformer Block
Feed Forward Neural Network
Multi-Head Self-Attention
Linear Projection + Position Embedding
Generate Image Patches

Figure 2.2: Architecture of a Vision Transformer for image classification. The input image is divided into patches, which are then linearly projected into patch embeddings. Positional embeddings are added to preserve spatial information. The transformer encoder processes these embeddings through multiple transformer blocks. Each block consists of a multi-head self-attention layer to capture long-range dependencies and a feed-forward neural network for further processing. A final classification layer processes the encoded patch embeddings to produce the classification output. Reprinted from Cameron [107]

2.2. Adversarial attacks

This section introduces the fundamental concepts and notations used throughout the paper regarding adversarial attacks. We focus on attacks against deep neural networks, denoted by $f_\theta(x)$, where θ represents the model's parameters and $x \in \mathcal{X}$ is the input data, belonging to the input space $\mathcal{X} \subset \mathbb{R}^d$. The output of the DNN is a prediction vector $f_\theta(x) \in \mathbb{R}^K$, where K is the number of classes. We assume a classification setting where the predicted class is given by $\arg\max_k f_\theta(x)_k$.

2.2.1. Adversarial Examples

An adversarial example is a carefully crafted perturbation, δ, of a benign input x, resulting in $x' = x + \delta$. This perturbation is designed to be imperceptible to humans, thus constrained in magnitude, $|\delta|_p \leq \epsilon$, while simultaneously causing the DNN, f_θ, to misclassify the input (as visualized in Figure 2.3). Formally, an adversarial example x' satisfies:

$$\begin{aligned} f_\theta(x') \neq f_\theta(x) \quad & \text{(Misclassification)} \\ ||\delta||_p < \epsilon \quad & \text{(Perceptual similarity)}, \end{aligned}$$

where $|\cdot|_p$ denotes the ℓ_p-norm, commonly ℓ_∞ (maximum perturbation per feature), ℓ_2 (Euclidean distance), or ℓ_1 (Manhattan distance), and ϵ represents the maximum permissible perturbation magnitude (the perturbation budget). Effectively, generating an adversarial example involves maximizing the loss function of the classifier, f_θ (or similarly, minimizing the classifier's confidence in the true label):

$$\delta = \arg\max_{||\delta|| \leq \epsilon} \mathcal{L}(f_\theta(x + \delta), \hat{y}),$$

where $\mathcal{L}$ is the loss function and $\hat{y}$ is the true label of x.

2.2.2. Representative attacks

Several methods exist for crafting adversarial examples. Here are a few representative attacks:

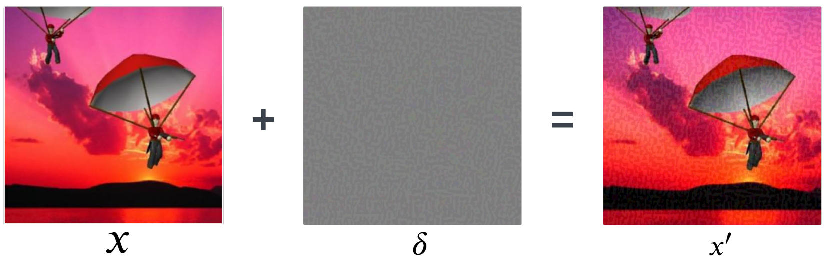


Figure 2.3: Illustration of adversarial example generation. The original image x is perturbed by δ to create the adversarial example x', which is misclassified by the DNN f_θ.

Fast Gradient Sign Method (FGSM): FGSM [36] is a fast, single-step attack that perturbs the input in the direction of the gradient of the loss function with respect to the input:

$$\delta = \epsilon \operatorname{sign}\left(\nabla_x L(f_\theta(x), \hat{y})\right)$$

where $sign(\cdot)$ is the sign function.

Projected Gradient Descent (PGD): PGD [65] is an iterative attack that extends FGSM. It iteratively applies FGSM, projecting the perturbed image back onto the ℓ_p-ball after each step:

$$x^{(t+1)} = \Pi_{x+S}\left(x^{(t)} + \alpha \operatorname{sign}\left(\nabla_x L(f_\theta(x^{(t)}), \hat{y})\right)\right)$$

where $x^{(0)} = x$, Π_{x+S} is the projection onto the set $S = \{\delta \mid \|\delta\|_p < \epsilon\}$, α is the step size and t is the iteration number.

Carlini & Wagner (C&W): The C&W attack [13] formulates adversarial example generation as an optimization problem, minimizing the distance between the adversarial example and the original input while ensuring misclassification:

$$\text{minimize } \|\delta\|_p + c \cdot f(x + \delta)$$
$$\text{subject to } x + \delta \in [0, 1]^d$$

where $f(\cdot)$ is a function designed to be negative when the model misclassifies and positive otherwise, and c is a hyperparameter controlling the trade-off between perturbation size and misclassification confidence.

These are just a few examples and many other attack methods exist.

2.3. Adversarial training

Adversarial training is a key defense strategy against adversarial attacks. It augments the training process by including adversarial examples in the training data. The core idea is to minimize the loss on both clean and adversarial examples. The most common form, min-max adversarial training, solves the following min-max optimization problem:

$$\min_\theta \mathbb{E}_{(x,\hat{y})\sim\mathcal{D}}\left[\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(f_\theta(x+\delta), \hat{y})\right]$$

where $\mathcal{D}$ is the training data distribution. This formulation aims to find model parameters θ that are robust to adversarial perturbations within the specified norm bound. The inner maximization problem finds the worst-case adversarial perturbation for a given input and model, while the outer minimization problem finds model parameters that minimize the loss on these worst-case examples. Pioneered by Goodfellow et al. [36] with FGSM, adversarial training was formalized as robust optimization by Madry et al. [65] using PGD, a stronger but computationally expensive attack. This cost motivated exploring more efficient methods, including revisiting FGSM, though it suffers from catastrophic overfitting [113], addressed by techniques like GradAlign [2].

The robustness-accuracy trade-off is another challenge of AT. TRADES [126] balances standard loss with a smoothness promoting term. CAT [12] uses a curriculum of increasing attack strengths. Other advancements include using FOSC [111] to assess adversarial example quality, FRL [120] for fairness, and OAT/OATS [109] for efficient robustness-accuracy trade-off exploration. These developments highlight the ongoing progress in adversarial training, addressing key challenges and providing a foundation for further research.

2.4. Adversarial purification

Adversarial purification (AP) offers an alternative defense strategy by introducing a separate purification module g_γ. This module acts as a pre-processing step, transforming the potentially adversarial input x' before it is fed to the classifier f_θ. The goal is to mitigate the impact of the adversarial perturbation δ, ideally resulting in the same classification output as the clean input: $f_\theta(g_\gamma(x+\delta)) = f_\theta(x)$. Notably, AP does not necessitate perfect reconstruction of the original input $(g_\gamma(x+\delta) \neq x)$. Instead, it focuses on removing the adversarial noise sufficient for correct classification. By focusing on removing classification-disrupting noise, rather than perfectly reconstructing the input, AP becomes a versatile tool applicable to a wide range of classifiers. This characteristic allows AP to function as a plug-and-play module, compatible with various classifiers and often implemented using pre-trained generative models for g_γ. The effectiveness of AP, however, relies heavily on the purifier's ability to distinguish and neutralize adversarial perturbations without excessively affecting the underlying semantic content of the input.

Initial purification methods leveraged generative models. Defense-GAN [86] projected inputs onto a GAN's range, but its effectiveness was tied to GAN quality. Later, Defense-VAE [57] used a VAE for faster purification. Self-supervised methods like NRP [72] trained a purifier to minimize feature distortion based on a fixed feature extractor. ZeroPur [10] combined readily available classifier features and blurring for a training-free approach. Diffusion models became central to AP, with DiffPure [75] utilizing forward and reverse diffusion. Enhanced versions like AGDM [60] incorporated guidance from an adversarially trained network for improved performance. Energy-based models offered an alternative, with ADP [123] employing a DSM-trained EBM for faster purification. Finally, AToP [61] combined adversarial training with purification, aiming to improve robustness and generalization.

2.5. Diffusion models

Diffusion models are a class of generative models that learn to synthesize data by iteratively denoising a sample from a known distribution. This process can be understood as the reverse of a diffusion process, which gradually adds noise to the data. We describe the forward and backward diffusion processes below. The neural network architecture typically used to parameterize the reverse process, U-Net [83], is detailed in Section 2.5.3.

2.5.1. Forward diffusion process

The forward diffusion process, also known as the diffusion process, gradually adds Gaussian noise to the data distribution $p(\mathbf{x}_0)$ over T time steps. This process can be defined by a Markov chain with transition probabilities $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ given by:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{2.1}$$

where $\beta_t \in (0,1)$ are variance schedules controlling the amount of noise added at each time step t, and $\mathbf{I}$ is the identity matrix. Commonly used schedules include linear and cosine schedules [73]. This formulation allows us to sample $\mathbf{x}_t$ at any time step t directly from $\mathbf{x}_0$ using:

$$p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2.2}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$. As t increases, $\mathbf{x}_t$ becomes increasingly noisy and converges to a standard Gaussian distribution when $T \to \infty$.

2.5.2. Backward diffusion process

The backward diffusion process, also known as the reverse process, aims to learn the reverse of the forward diffusion. It starts from a sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively removes noise to generate a sample from the data distribution $p(\mathbf{x}_0)$. The reverse process is also a Markov chain defined by

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\gamma(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\gamma(\mathbf{x}_t, t)), \tag{2.3}$$

where $\boldsymbol{\mu}_\gamma(\mathbf{x}_t, t)$ and $\boldsymbol{\Sigma}_\gamma(\mathbf{x}_t, t)$ are the learned mean and covariance, respectively, parameterized by a neural network with parameters γ. As described in Section 2.5.3, this neural network is typically a U-Net architecture [22, 91]. In practice, the covariance is often fixed to a time-dependent constant derived from the forward process variance schedule, and only the mean is learned. The goal of training a diffusion model is to learn the parameters γ such that the reverse process faithfully approximates the true posterior $p(\mathbf{x}_{t-1}|\mathbf{x}_t)$. This is typically done by minimizing a variational bound on the negative log-likelihood.

2.5.3. U-Net

The U-Net [83], originally designed for biomedical image segmentation, has become essential for diffusion-based generative models like Denoising Diffusion Probabilistic Models (DDPMs). Its strength lies in capturing both high-level context and fine-grained details, making it ideal for detailed image reconstruction.

The U-Net works as an encoder-decoder with skip connections. The encoder downsamples the input image, extracting increasingly abstract features through convolutions and pooling, capturing the global context. The decoder then upsamples these features back to the original resolution, reconstructing the image's spatial structure. Skip connections link corresponding encoder and decoder layers, preserving fine details lost during downsampling.In DDPMs, the U-Net forms the core of the denoising network. This network iteratively predicts the noise in an image during the reverse diffusion process. At each step, the U-Net receives a noisy image. It predicts the noise added at that step and subtracts it, progressively refining the image back to its clean version.The U-Net's encoder captures high-level features. These features are passed to the decoder, which reconstructs the clean image using both the global context and the detailed information from the skip connections. These skip connections are essential in DDPMs, ensuring the network retains fine details from the noisy image for accurate reconstruction. DDPMs enhance the U-Net with mechanisms to handle temporal and spatial dependencies. Positional encodings inform the model about the current timestep in the diffusion process, allowing it to adapt its predictions based on the noise level. Attention mechanisms can also be incorporated to capture long-range dependencies, improving generated sample quality. Further architectural refinements, like scale-shift normalization in attention layers and residual blocks with up/down sampling, boost the model's ability to learn complex denoising distributions, leading to better generation quality and diversity.

By leveraging the U-Net's ability to model both global and local features, DDPMs excel at generating high-quality, diverse images [43]. See Figure 2.4 for a detailed illustration of the U-Net architecture.

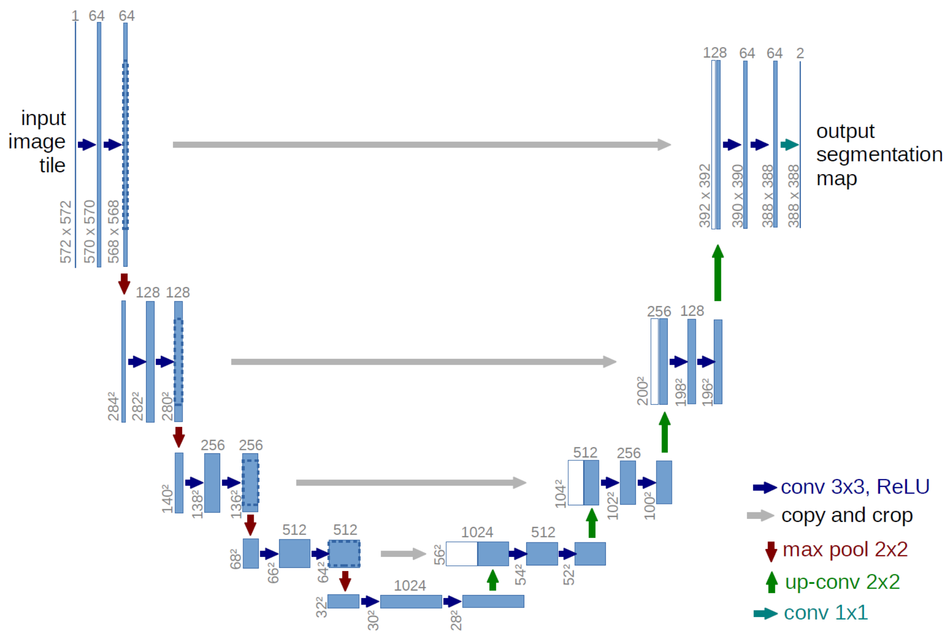


Figure 2.4: The U-Net architecture (illustrated for a 32x32 pixel resolution at the lowest level) features blue boxes representing multi-channel feature maps. The number of channels is indicated above each box, while the dimensions of the feature map (x and y size) are shown at the bottom-left corner. White boxes signify copied feature maps, and the arrows indicate various operations within the network. Adapted from Ronneberger et al. [83]

2.6. Measuring standard and robust accuracy

Evaluating the effectiveness of defenses against adversarial attacks requires measuring both standard accuracy (performance on clean data) and robust accuracy (performance on adversarially perturbed data). These metrics provide a comprehensive assessment of a model's ability to generalize to unseen data and withstand adversarial attacks.

2.6.1. Standard Accuracy

This is the conventional accuracy metric, calculated as the percentage of correctly classified clean samples:

$$\text{Standard Accuracy} = \frac{\text{Number of correctly classified clean samples}}{\text{Total number of clean samples}} * 100 \tag{2.4}$$

2.6.2. Robust Accuracy

This measures the model's robustness to adversarial attacks. It is computed as the percentage of correctly classified adversarial examples:

$$\text{Robust Accuracy} = \frac{\text{Number of correctly classified adversarial examples}}{\text{Total number of adversarial examples}} * 100 \tag{2.5}$$

In evaluating defenses, it's important to report both standard and robust accuracy to understand the potential trade-off between these two metrics. Ideally, a robust defense should maintain high standard accuracy while considerably improving robust accuracy compared to undefended models.

3 Literature review

3.1. The growing threat of adversarial attacks

Adversarial attacks pose a threat to DNNs by exploiting their vulnerabilities. These attacks subtly perturb the input data, often imperceptibly, leading to misclassifications [99]. We can broadly classify these attacks into white-box attacks, where the attacker possesses full knowledge of the target model and black-box attacks, where the attacker can only query the model and observe its outputs. Historically, adversarial attacks were predominantly white-box. However, the field has changed with the emergence of sophisticated black-box techniques, thereby extending the scope of the threat. These attacks can be further distinguished as either targeted, which aims to induce a specific misclassification or untargeted which aims for any incorrect classification.

This increasing sophistication of adversarial attacks naturally raises concerns regarding the security of DNNs across various applications, especially in safety-critical domains such as autonomous driving. Consider, for instance, the scenario in which an image classification model within a self-driving vehicle misclassifies a stop sign as a speed limit sign due to adversarial manipulation. The development of universal adversarial perturbations (UAPs) [70, 69, 58], which are capable of fooling a model on a wide range of inputs, further amplifies this concern. As these attacks become more powerful and easier to execute, developing robust defenses against them is critical to ensuring the safe and reliable deployment of DNNs.

3.2. The importance of robust defenses

As DNNs become widespread in various aspects of our lives, the importance of ensuring their reliability and security cannot be underestimated. The vulnerability of these systems to adversarial attacks poses a substantial risk to their integrity. Adversarial attacks can have far-reaching consequences, ranging from compromising the safety of autonomous vehicles to undermining the fairness of decision-making systems [130]. Therefore, it is vital to develop effective defenses that can protect DNNs against such attacks. The need for robust defense methods is further stressed by the fact that adversarial attacks are increasingly sophisticated and accessible [18].

The rise of tools that provide techniques for generating adversarial examples, such as Foolbox [81], Advertorch [23] or Adversarial Robustness Toolbox (ART) [74], has made it easier for attackers to craft attacks. Moreover, the growing use of DNNs in high-stakes applications, such as healthcare and finance, has created a need for defense mechanisms that can ensure the reliability and security of these systems [114].

In this context, the development of effective defense strategies against adversarial attacks is a necessity. Despite the growing recognition of the importance of defending against adversarial attacks, the development of effective defense mechanisms remains a challenging task. Adversarial attacks are often designed to exploit the specific vulnerabilities of DNNs, making it difficult to develop defense mechanisms that can provide comprehensive protection [18]. In addition, the arms race between attackers and defenders is ongoing, with new attack techniques and defense strategies continuously emerging.

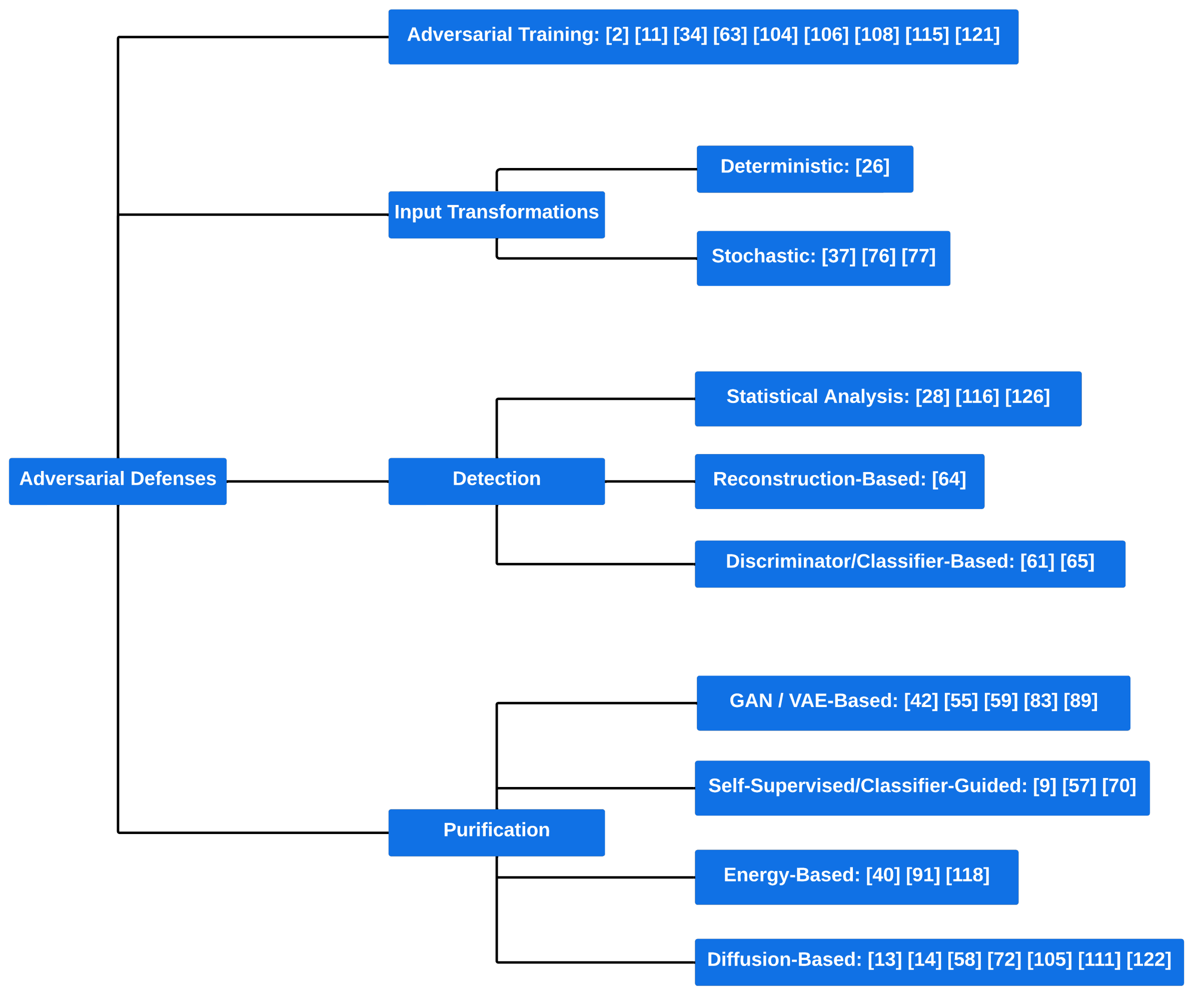

Adversarial Training: [2] [11] [34] [63] [104] [106] [108] [115] [121]
Input Transformations
Deterministic: [26]
Stochastic: [37] [76] [77]
Adversarial Defenses
Detection
Statistical Analysis: [28] [116] [126]
Reconstruction-Based: [64]
Discriminator/Classifier-Based: [61] [65]
Purification
GAN / VAE-Based: [42] [55] [59] [83] [89]
Self-Supervised/Classifier-Guided: [9] [57] [70]
Energy-Based: [40] [91] [118]
Diffusion-Based: [13] [14] [58] [72] [105] [111] [122]

Figure 3.1: This diagram categorises adversarial defenses, including adversarial training, input transformations, detection, and purification methods. Each category is further divided into specific techniques with corresponding references (indicated by bracketed numbers). Adversarial training includes various methods. Input transformations can be deterministic or stochastic. Detection methods include statistical analysis, reconstruction-based approaches, and discriminator/classifier-based techniques. Purification methods leverage GANs/VAEs, self-supervised/classifier-guided learning, energy-based models, and diffusion-based approaches.

3.3. An overview of defense strategies

This section briefly covers two defense strategies against adversarial attacks: input transformations and detection methods. While we also consider adversarial training and purification as important defense strategies, a more detailed discussion of these techniques is reserved for later sections.

Input transformations: The development of robust defenses against adversarial examples has led to the investigation of input transformation as a purification technique. While this approach has shown promise, challenges persist. Initial efforts focused on leveraging readily available image processing techniques like JPEG compression [27], which removes adversarial perturbations by projecting the input onto a manifold of clean images. Similarly, bit-depth reduction and image cropping/rescaling were used to purify adversarial perturbations. However, these deterministic and often differentiable transformations proved inferior against stronger adversaries capable of exploiting obfuscated gradients [5]. This proved the need for defenses that are both non-differentiable and stochastic.

The introduction of randomized resizing and padding represented a turning point towards stochastic defenses, demonstrating improved robustness, particularly against iterative attacks. However, the lack of a theoretical foundation and limitations in effectiveness motivated the exploration of more sophisticated methods [119]. Guo et al. [39] investigated total variance minimization and image quilting, leveraging their non-differentiable nature for a more resilient defense. Yet, computational complexity and parameter sensitivity came into view as new challengers. Recognizing the limitations of single transformations, Raff et al. [80] pioneered the concept of ensembling multiple weak transformations with their BaRT defense. This stochastic combination of diverse transformations significantly improved robustness but still faced challenges related to heuristic selection and the computational cost of BPDA+EOT. Building upon this insight, Qiu et al. [79] proposed a two-step transformation incorporating DCT-based quantization and pixel dropping/displacement, demonstrating the effectiveness of designing specialized, non-approximable transformations. This progression reveals a clear trend towards more complex and stochastic transformations for purification, but the search for theoretically grounded, computationally efficient, and truly robust defenses continues.

Detection Methods: Early work, such as that by Feinman et al. [29], explored implicit detection by leveraging artifacts of adversarial manipulation. They analysed density estimates in the feature space and Bayesian uncertainty derived from dropout networks, suggesting that adversarial examples reside off the data manifold and in low-confidence regions. While promising initial results were observed, this approach's reliance on dropout limited its applicability to specific network architectures, and the effectiveness against adaptive attacks remained an open question. Similarly, Xu et al. [121] introduced feature squeezing, a detection technique based on comparing predictions on the original input and a simplified, squeezed version. This method shows simplicity and computational efficiency, but its vulnerability to adaptive attacks highlighted an essential limitation of the defense.

Addressing the need for stronger defenses, Lu et al. proposed SafetyNet [63], which utilizes quantized ReLU activations and a Support Vector Machine (SVM) to detect anomalous activation patterns showing adversarial examples. The discrete nature of the quantized activations created a harder optimization problem for the adversary, enhancing robustness against both seen and unseen attacks. However, SafetyNet's dependence on a specific network architecture limited its generalizability and the potential for approximation through smoothing remained a concern. Metzen et al. [67] took a different approach, augmenting the classifier with a ”detector” subnetwork trained to discriminate between genuine and adversarial inputs. This approach demonstrated high detection accuracy and some generalization to weaker adversaries. However, it is susceptible to a ”dynamic adversary” that attacks both classifier and detector concurrently. Meng & Chen [66] further explored detection by modeling the manifold of normal data using autoencoders. This allowed for both detection, based on reconstruction error or probability divergence and change projecting adversarial examples back onto the manifold. Although MagNet shows promising results against black-box attacks, it proved vulnerable to white-box attacks. Finally, Zheng & Hong. [131] focused on the internal representations of the network with their framework I-defender, using Gaussian Mixture Models to characterize the expected distribution of hidden neuron activations. This method achieved state-of-the-art performance in unsupervised detection, but its dependence on fully connected layers and vulnerability to white-box attacks highlighted the need for further refinement of methods that analyse the network's internal state.

Collectively, these works illustrate the evolution of adversarial example detection techniques, moving from detecting artifacts of specific attacks towards characterising expected network behavior and identifying deviations. Despite significant advancements, the limitations of existing methods, particularly their vulnerability to adaptive attacks and the reliance on specific network architectures or training data, motivate ongoing research into more robust and generalizable detection and purification strategies. A promising direction involves developing defenses that are less dependent on specific attack characteristics and more focused on understanding and enforcing the intrinsic properties of benign data and network behavior.

3.4. From basic adversarial training to advanced techniques

The core principle of adversarial training, as initially explored by Goodfellow et al. [36], involves augmenting the training data with adversarial examples, forcing the model to learn to correctly classify both clean and perturbed inputs. Goodfellow et al. also proposed the Fast Gradient Sign Method (FGSM), a single-step method for generating adversarial examples and suggested that the linearity of deep networks contributes to their vulnerability to these attacks. This work laid the groundwork for later research on adversarial training.

Madry et al. [65] innovated the field by formalizing adversarial training as a robust optimization problem, framing it as a min-max game between the model and an adversary. They proposed using stronger iterative attacks like Projected Gradient Descent (PGD) to generate adversarial examples, achieving state-of-the-art robustness. Their work highlighted the importance of both the attack strength and model capacity in identifying the effectiveness of adversarial training. However, this PGD-based approach, while effective, suffers from computational overhead due to the iterative nature of the attack.

This computational bottleneck stimulated research into more efficient adversarial training methods. One direction was to revisit FGSM-based training due to its speed. However, as highlighted by Wong et al. [113] and further investigated by Andriushchenko and Flammarion [2], FGSM training is prone to a phenomenon called catastrophic overfitting (CO), where the model's robustness against multi-step attacks collapses abruptly during training, despite maintaining high accuracy on single-step adversarial examples. Andriushchenko and Flammarion delved deeper into the mechanisms of CO, demonstrating its occurrence even in single-layer networks and attributing it to the development of local non-linearity. They proposed GradAlign [2], a regularization technique based on maximizing gradient alignment within the perturbation set, to mitigate CO and improve the performance of FGSM training.

Another significant challenge in adversarial training is the inherent trade-off between robustness and standard accuracy. Zhang et al. addressed this trade-off directly by introducing TRADES [126], an algorithm that formulates adversarial training as the optimization of a regularized surrogate loss function. TRADES explicitly balances the standard classification loss with a term promoting smoothness around the decision boundary, effectively trading off between clean accuracy and robustness against adversarial perturbations. Furthermore, Cai et al. observing the limitations of standard adversarial training on complex datasets, proposed Curriculum Adversarial Training (CAT) [12], which employs a curriculum of increasingly strong attacks. CAT, along with techniques like batch mixing and quantization to combat catastrophic forgetting and improve generalization, achieved notable improvements in robustness.

Wang et al. realising the important role of the quality of adversarial examples used in training, introduced the First-Order Stationary Condition (FOSC) [111] as a metric for evaluating the convergence quality of these examples. They showed that while high-quality (low FOSC) examples are important for robustness in later training stages, they may be detrimental in early stages. This insight led to a dynamic training strategy that gradually increases the required convergence quality of adversarial examples, resulting in improved robustness. Simultaneously, Xu et al. brought to light the issue of fairness in adversarial training, revealing significant difference in robustness and accuracy across different classes or groups, even in balanced datasets. They proposed the Fair Robust Learning (FRL) framework [120], adding fairness constraints and dynamically adjusting perturbation margins during training, to mitigate these difference in robustness.

The computational overhead of retraining models multiple times to explore different points on the robustness-accuracy trade-off curve was addressed by Wang et al. with Once-for-all Adversarial Training (OAT) [109]. OAT allows for the in-situ calibration of this trade-off at test time by treating the robust loss weight as a model input. By employing dual batch normalization, OAT effectively handles the conflicting feature statistics of clean and adversarial examples, enabling a single model to achieve a range of robustness-accuracy trade-offs without retraining. This framework was further extended to OATS, which incorporates model slimming for joint optimization of robustness, accuracy and model complexity.

Despite significant advancements in adversarial training, several limitations persist. The computational cost of robust training, particularly with iterative attacks like PGD, remains a major obstacle. Even fast, single-step methods like FGSM are susceptible to catastrophic overfitting. Furthermore, adversarial training often necessitates a trade-off between robust and standard accuracy, and achieving an optimal balance can be challenging. Existing methods, while attempting to address this trade-off, still require careful hyperparameter tuning and may not fully capture the complexities of the trade-off space. Another limitation is the potential for unfairness, where adversarial robustness is not evenly distributed across different classes or groups within the data. Finally, efficiently exploring the robustness-accuracy trade-off often requires multiple retraining runs, adding to the computational overhead. Although techniques like OAT offer in-situ calibration, they still rely on careful selection of training hyperparameters and model configurations.

3.5. Introduction to purification

Adversarial purification offers a unique proactive defense against adversarial examples, destroying adversarial perturbations before classification. While other defenses aim to mitigate adversarial perturbations, purification differentiates itself by actively reshaping potentially adversarial inputs to match benign data points, exploiting the underlying structure of the data manifold. In contrast to adversarial training, which enhances model robustness by introducing adversarial examples during the training process, purification effectively ”cleanses” or projects perturbed inputs back onto the data manifold prior to classification. This approach also contrasts with input transformation defenses, which pre-process inputs to filter adversarial perturbations and adversarial example detection methods, which identify and flag such inputs without modification. Rather than relying on adversarial samples, input transformations or explicit detection criteria, purification leverages the inherent structure of the data distribution to neutralize adversarial perturbations before they are fed to the classifier.

3.6. Adversarial purification methods

The search for robust deep learning models that can withstand adversarial attacks has driven significant advancements in defense strategies, with purification methods playing a central role. This review summarizes and evaluates key papers in the field, highlighting the evolution from simple adversarial training to advanced purification techniques using generative models, self-supervision, and certified defenses.

Adversarial Training: Initial defenses focused on adversarial training. For instance Madry et al. [65] augmented training data with adversarial examples. While effective against known attacks, as demonstrated by its robustness against PGD attacks in Hill et al.’s study [42], AT exhibits limitations. It often overfits to the training attacks, weak against unseen threats, as highlighted by Lin et al.’s AToP paper [61] and the analysis of unseen attacks in multiple studies [53, 24]. Furthermore, AT is computationally expensive and can significantly degrade standard accuracy, issues addressed by Naseer et al.’s NRP [72].

Early Purification: Defense-GAN [86], pioneered purification using generative models. By projecting inputs onto the range of a pre-trained GAN’s generator prior, it defended against both black-box and white-box attacks, addressing the attack-specificity of AT. However, its effectiveness depends much on GAN quality and requires iterative optimization, a computational bottleneck addressed by Defense-VAE [57]. Using a Variational autoencoder (VAE), Defense-VAE achieved faster, one-shot purification, making real-time defense feasible, but still relied on adversarial examples for training, introducing vulnerability to unseen attacks. PixelDefend [93] used a PixelCNN for purification, achieving attack-agnosticism and strong performance, but the computational cost of PixelCNN remained a limitation. Similarly, PuVAE [44] used a class-conditional VAE, improving on Defense-VAE with targeted denoising, but its reliance on a source classifier and vulnerability to unseen attacks are it’s limitations.

Self-Supervision and Classifier Guidance: Seeking more efficient and adaptable defenses, researchers invented self-supervised and classifier-guided purification techniques. HGD [59], for example, guided a denoising autoencoder using high-level feature representations extracted from the classifier itself, demonstrating improved robustness and generalization compared to pixel-level denoising. Park et al.’s ZeroPur [10] simplified purification further by combining readily available classifier features with a blurring transformation, eliminating the need for training a separate purification model. This training-free approach, while computationally efficient, proved susceptible to attacks specifically designed to be robust to blurring. NRP [72] took a different approach, training a dedicated purifier network to minimize the distortion of features extracted by a fixed, pre-trained network. This self-supervised approach achieved strong cross-task performance, but introduced a potential vulnerability due to the reliance on a fixed feature extractor, which could be exploited by an attacker with knowledge of that extractor’s characteristics.

Adversarial purification using diffusion models: Diffusion models have become central to adversarial purification. DiffPure [75] introduced diffusion-based purification, using the forward process to add noise and the reverse process for denoising. They employed the adjoint method for efficient gradient computation, enabling evaluation against strong adaptive attacks. GDMP [110] and AGDM [60] enhanced DiffPure with guidance, improving semantic preservation and robustness. AGDM's use of an adversarially-trained auxiliary network for guidance offered superior performance. DDS [14] and DiffSmooth [127] adapted diffusion for certified robustness within the denoised smoothing framework. DDS achieved state-of-the-art certified robustness at low perturbation levels but struggled with higher levels due to the diffusion model's tendency to hallucinate, an issue partially mitigated by DiffSmooth's local smoothing. DensePure [116] achieved improved certified robustness through multiple reverse diffusion runs and majority voting, approximating the highest density region in the conditional distribution. However, the computational cost of diffusion-based methods remains a key challenge. RDC [15] integrated the diffusion model directly into the classifier, demonstrating strong robustness and generalization as a generative classifier, but shared the computational limitations of diffusion models.

Adversarial purification with Energy-based models ADP [123] used a DSM-trained EBM and randomized purification, providing a faster alternative to diffusion-based purification with certified robustness. Hill et al. [42] advanced EBM-based defenses with convergent EBMs, enabling effective purification with long-run MCMC and achieving competitive performance with AT for standardly trained classifiers. MALADE [95] further refined this by incorporating a novel conditional gradient estimator, improving performance against strong attacks, but the computational cost of MCMC remains a limitation.

Adversarial Training of Purifiers AToP [61] combined adversarial training with purification, training the purifier with adversarial examples to improve robustness and generalization. This approach leverages the strengths of both AT and purification but inherits the computational cost associated with generative models.

Looking forward, several critical research directions stand out. Improving the computational efficiency of purification, especially for diffusion models, remains top priority. Extending certified robustness guarantees to more purification methods is equally important. Developing sophisticated, adaptive guidance mechanisms is a promising opportunity. Addressing vulnerabilities to advanced adaptive attacks, especially those targeting purification modules or exploiting specific model weaknesses. Exploring hybrid defenses combining purification with other techniques will significantly amplify the impact of this research domain. Figure 3.1 illustrates the existing field of defense mechanisms, including various purification approaches

3.7. Research gap

Despite significant progress in adversarial purification techniques, several research gaps remain, motivating this thesis. Existing methods, including those leveraging diffusion models, show trade-offs between standard accuracy, robustness against diverse attacks (especially color-based attacks) and computational efficiency.

Adversarial training, while offering strong robustness against known attacks, struggles with unseen attacks and often degrades standard accuracy [53, 101]. Purification methods using GANs, VAEs, and PixelCNNs have shown promise but face limitations in terms of GAN quality, computational cost or reliance on adversarial examples for training [93, 57, 86]. Even recent advances in diffusion-based purification like DiffPure, GDMP, and AGDM remain vulnerable to color-based attacks due to the sensitivity of DDPMs to chromatic manipulations [75, 110, 60]. Furthermore, achieving a balance between effective perturbation removal and preserving image semantics, particularly for large-scale datasets, remains a challenge [110].

Particularly, current diffusion-based purification methods lack a targeted approach to perturbation removal. They typically operate globally, applying denoising across the entire image, which can affect benign features and impact standard accuracy. Additionally, the reliance on global denoising makes them susceptible to attacks that manipulate subtle color information, which might be overlooked by global operations.

This research addresses these gaps by proposing MSID, a novel purification method that utilises a multi-scale inpainting approach guided by occlusion sensitivity maps. By targeting specific potentially perturbed regions for inpainting using a pre-trained DDPM, MSID aims to achieve a superior balance between robustness (including against color-based attacks) and standard accuracy compared to existing methods. This targeted approach also hypothesizes to offer better preservation of image semantics and improved generalization to unseen attacks.

In summary, the research gaps addressed by this thesis are:

- **Limited robustness against unseen attacks:** Existing adversarial training methods tend to overfit to known attacks.
- **Vulnerability to color-based attacks:** Current diffusion-based purification methods are susceptible to chromatic manipulations.
- **Trade-off between robustness and standard accuracy:** Achieving high robustness often comes at the cost of reduced standard accuracy.
- **Lack of targeted perturbation removal:** Global denoising in diffusion-based methods can affect benign image features.
- **Preserving image semantics:** Effectively removing perturbations while maintaining image fidelity remains a challenge.

MSID aims to bridge these gaps by introducing a targeted, multi-scale inpainting approach for adversarial purification using diffusion models.

4 Methodology

4.1. Motivation

This chapter outlines the methodology used in this thesis to address the vulnerability of deep learning models to adversarial attacks. The focus is on developing a robust AP defense, using DDPM. This approach is motivated by observations and limitations in existing defense strategies.

AT while effective in enhancing robustness, struggles to generalize to unseen attacks, often facing a trade-off between standard and robust accuracy. It is also computationally expensive and requires retraining the model for each new attack [53, 103]. These limitations highlight the need for alternative approaches, such as AP, which can generalize better and be applied to pre-trained models.

Adversarial purification is designed to ”cleanse” adversarial examples by projecting them back onto the manifold of clean images, effectively removing adversarial perturbations before inference. Recent advancements in generative models, particularly DDPMs, demonstrate their ability to generate high-quality, realistic images [22]. This capability makes DDPMs a promising tool for adversarial purification, as their data distribution modeling can effectively neutralize adversarial noise [75].

Existing AP methods often purify the entire image, potentially altering benign features. To address this issue, this work uses XAI techniques, specifically multi-scale occlusion sensitivity maps, to identify regions of the image potentially adversarially perturbed. This enables targeted purification, focusing on removing adversarial noise only from specific areas while preserving other image features.

Using DDPMs for inpainting presents challenges, as masking sensitive regions can limit the model of necessary context for accurate reconstruction. To overcome this, strategies like jittered grid unmasking and VPS artifact removal are developed to ensure the inpainting process produces realistic, artifact-free images.

The methodology proposed in this thesis, termed Multi-Scale Inpainting Defense (MSID), leverages the generative power of DDPMs for targeted and context-aware adversarial purification. By integrating DDPM-based inpainting with multi-scale occlusion sensitivity maps, this approach intends to create a robust defense mechanism that addresses the limitations of existing fmethods. It neutralises adversarial perturbations effectively while preserving benign features of the image and model accuracy, solving the core challenges of robustness and generalization. This novel application of generative models and XAI techniques offers a promising solution for adversarial defense.

4.2. Our defense

Our defense method consists two main stages (see Figure 4.1): (1) generating occlusion sensitivity maps and (2) restoring image via an inpainting diffusion model. The first stage leverages the observation that different image regions, at varying scales, contribute differently to a class’ classification score (see **(b)** in 4.1 and Appendix C for examples across scales). Analyzing the occlusion sensitivity map of an adversarial example reveals the region most responsible for misclassification. The second stage

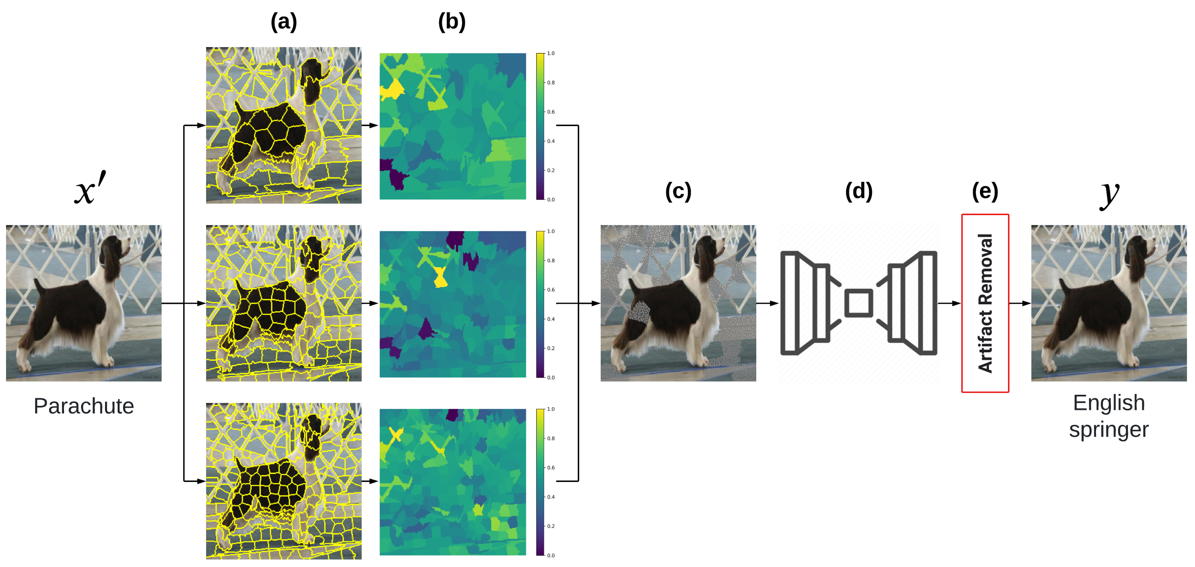


Figure 4.1: The Multi-Scale Superpixel Inpainting for Defense (MSID) method removes targeted adversarial perturbations from images in four steps. First **(a)**, the adversarial example x' undergoes multi-scale superpixel segmentation, capturing both coarse and fine details to precisely identify potentially perturbed regions. Second **(b)**, occlusion sensitivity maps are generated at each superpixel scale. This involves blurring individual superpixels and measuring the resulting impact on classifier output, effectively highlighting potentially perturbed areas. Third **(c)**, based on these sensitivity maps, a mask M is created to identify highly sensitive regions, which are then restored using a pre-trained DDPM **(d)**. This targeted inpainting removes the perturbations while preserving uncorrupted image features. Finally **(e)**, Variance Preservation Sampling (VPS) is applied to the restored image to mitigate grid-based inpainting artifacts, ensuring an artifact-free final output y.

then inpaints these sensitive, likely perturbed, regions using a DDPM-based inpainting model. This simultaneously removes the adversarial perturbations while preserving image similarity to the benign one.

4.3. Generate occlusion sensitivity maps

XAI research primarily aims to reveal the underlying reasoning in machine learning models [54]. Numerous methods have been introduced, including attribution techniques [56], concept-based approaches [33, 48], and global analysis tools [35, 4]. In our approach, we applied a model-agnostic technique that focuses on occluding specific features and observing the resulting changes in model predictions [125]. Occlusion sensitivity offers several advantages as an XAI method, particularly regarding its simplicity and speed [30, 106] Two key factors determine the accuracy of an occlusion sensitivity map: pixel grouping and the choice of imputer for occluded features. These choices are critical because they directly affect the map's ability to identify true sensitive regions and therefore, the effectiveness of any defense built upon it. Incorrect choices can lead to a weak defense. To ensure accurate identification of sensitive regions, we use superpixels for grouping, which allows us to capture meaningful image features. Furthermore, we use Gaussian blurring technique to impute occluded superpixels. Below we detail the rationale behind these choices and explain why they are critical for building a robust defense.

4.3.1. Pixel grouping

A core component of MSID is its pixel grouping strategy. Knowing that individual pixels in an image $x \in \mathbb{R}^{w \times h \times n}$, with width w, height h, and n color channels often represent redundant information due to their proximity [11], MSID uses superpixels – clusters of pixels treated as singular features. This reduces computational complexity, as the number of superpixels $n_{superpixels} < w \cdot h \cdot n$. Specifically, we use the Simple Linear Iterative Clustering (SLIC) algorithm [1] to segment the image into superpixels, denoted as $N = \{1, 2, \ldots, n_{superpixels}\}$. We chose to use SLIC, because it generates superpixels that adhere to local gradients in the image, effectively grouping contextually related pixels [97]. Figure 4.2 shows an example of an image segmented into superpixels using SLIC. This not only improves computational efficiency but also enhances the interpretability of the sensitivity maps used later to

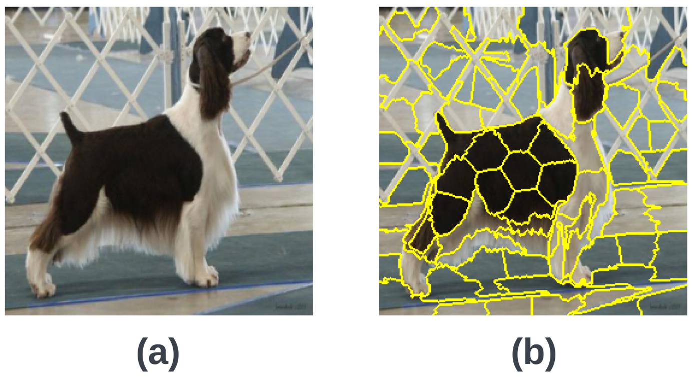


Figure 4.2: Example of superpixel segmentation using the SLIC algorithm. Image **(a)** shows the original image, and image **(b)** shows the resulting superpixel segmentation.

identify regions that contain adversarial perturbations. By analysing these superpixel-based sensitivity maps, we gain a clearer understanding of how coherent image features contribute to the classifier's decisions, forming the foundation for the subsequent feature occlusion and multi-scale analysis within MSID.

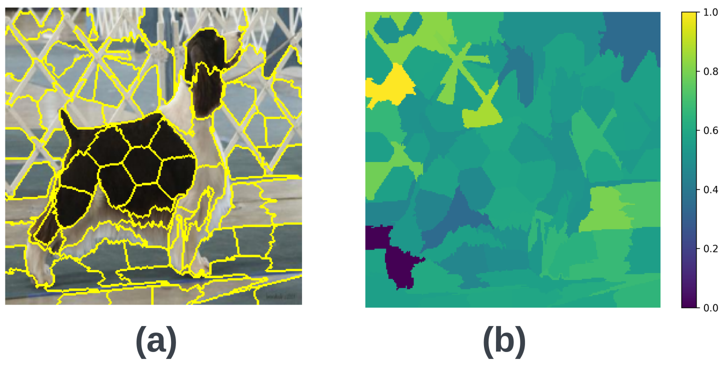


Figure 4.3: Example of a sensitivity map. **(a)** Original image segmented into superpixels. **(b)** Sensitivity map generated via feature occlusion, where each superpixel's sensitivity is represented. Brighter regions indicate higher sensitivity to occlusion, meaning these areas are more important for the classifier's decision.

4.3.2. Feature occlusion

Feature occlusion plays an important role in understanding the classifier's decision-making process and guiding our defense strategy. This technique involves systematically masking portions of the image, specifically superpixels, to analyse their influence on the classifier's predictions. By observing the changes in output when different regions are occluded, we can identify the most sensitive areas, those that have the greatest impact on classification.

In creating the occluded prediction $f_c(x_S)$ (where x_S denotes the part of the image X restricted to the feature subset S, identified using SLIC), it is generally infeasible to exclude features $\overline{S}$ (the complement of S) entirely. Instead, their influence on the model's predictions must be minimized. The only model-agnostic approach is to generate occluded samples $(x_S, X_{\overline{S}})$, where $X_{\overline{S}}$ represents artificially generated values provided by an imputer q. The occluded model prediction is then expressed as:

$$f_c(x_S) = \sum_{X_{\overline{S}} \sim q} f_c(x_S, X_{\overline{S}}).$$

Here, we used the marginal distribution $q = p(X_{\overline{S}})$ to decouple S and $\overline{S}$. Our chosen imputer, a

Gaussian blurring technique, replaces occluded regions with a blurred effect, preserving the broader structure of the image while eliminating finer details. Given a binary mask M, the blurred image X' is defined as:

$$X' = \text{Blur}(X_{\overline{S}}, M, \sigma)$$

The Gaussian blur, controlled by its standard deviation σ, is chosen as our imputer because it hides, rather than erases, features. This helps minimize out-of-distribution (OOD) artifacts by maintaining contextual similarity in occluded areas [31]. Furthermore, blurring can bring adversarial examples closer to their ground-truth labels, potentially alleviating adversarial properties [10].

The result is a sensitivity map, highlighting the regions that, when occluded cause the change in the classifier's decision. Figure 4.3 shows an example of a sensitivity map. These high-sensitivity areas are critical for classification, and are thus likely targets for identifying adversarial perturbations. Importantly, feature occlusion is a model-agnostic technique, requiring no knowledge of the classifier's internal workings. This allows us to analyse any black-box classifier without access to its architecture or parameters. The generated sensitivity map allows us to focus our restoration on the most sensitive regions of the image.

4.3.3. Multi-scale superpixel segmentation

Traditional occlusion-based explanation methods rely on occluding fixed-size patches across the image [45]. However, this approach fails to consider the hierarchical nature of image features, where both fine details and larger structures contribute to the overall classification. To capture the importance of both fine details and larger structures, we introduce multi-scale superpixel segmentation. Figure 4.1(a-b) illustrates the process of multi-scale superpixel segmentation and the resulting sensitivity maps. This approach analyses the image at multiple levels of granularity, providing a richer understanding of feature contributions to the classification.

Instead of using a single superpixel segmentation scale, we generate superpixels at three scales (i.e. fine, medium, and coarse). This allows us to capture hierarchical information, understanding that fine-grained details like textures may be essential for some classifications, while broader structures like shapes are important for others. For instance, distinguishing between dog breeds might rely on subtle texture variations captured at a fine scale, whereas differentiating a dog from a cat might depend on the overall shape identified at a coarser scale.

This multi-scale analysis results in more informative sensitivity maps. These maps reflect the hierarchical contributions of image features, providing a more subtle understanding of regional sensitivity. This approach allows more precise targeting for our defense mechanism.

Finally, the multi-scale sensitivity maps guide the subsequent inpainting process. Knowing which regions are important at different scales allows the inpainting model to restore these regions, ensuring the preservation of overall image structure. This multi-scale approach facilitates more accurate and contextually appropriate image restoration, further enhancing the robustness of our defense.

4.3.4. Map fusion

We combine sensitivity maps from multiple scales using a weighted average, prioritizing coarser maps to highlight major regions while incorporating finer details. Coarser maps, with their broader view, provide the overall structure, while finer maps add the necessary precision. The fused sensitivity map is generated as follows: First, distinct regions are identified within the coarsest-scale sensitivity map. Then, for each identified region, a weighted average of the corresponding areas in the finer-scale maps is calculated (see Equation (4.1)). The weights w_n, where $n \in 1, 2, 3$, correspond to the relative importance of scales 1, 2, and 3, respectively.

The term F_n denote the set of elements associated with n-th fine scale, while $|F_n|$ indicates the number of elements in this set. Within the summations, S_j represents the score corresponding to each superpixel j in the respective sets. Finally, S_1 is the score for coarse scale. This iterative masking and averaging process yields a smoothed sensitivity map, highlighting the most important areas with

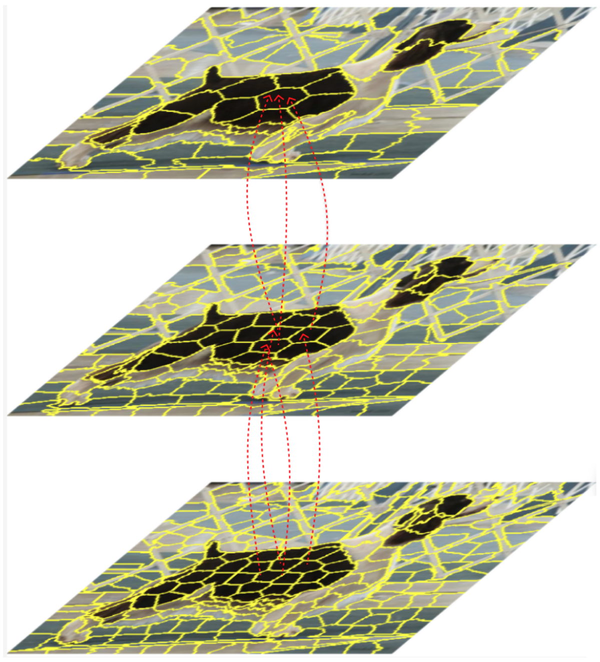


Figure 4.4: Visualization of the multi-scale sensitivity map fusion process. The arrows indicate how regions in finer-scale maps correspond to coarser maps. The resulting combined map integrates information from all scales according to Equation 4.1.

increased accuracy (see equation 4.1). Figure 4.4 illustrates this fusion process, showing how information from different scales is combined.

$$S_{\text{combined}} = w_3 \cdot \left(\frac{1}{|F_3|} \sum_{f \in F_3} S_j \right) + w_2 \cdot \left(\frac{1}{|F_2|} \sum_{f \in F_2} S_j \right) + w_1 \cdot S_1 \tag{4.1}$$

Our multi-scale approach offers a significant advantage by integrating hierarchical spatial information, providing a deeper understanding of regional significance at different levels. This results in a refined heatmap that accurately reflects important image areas without reducing detail, thus offering richer insights into the classifier's decision-making process.

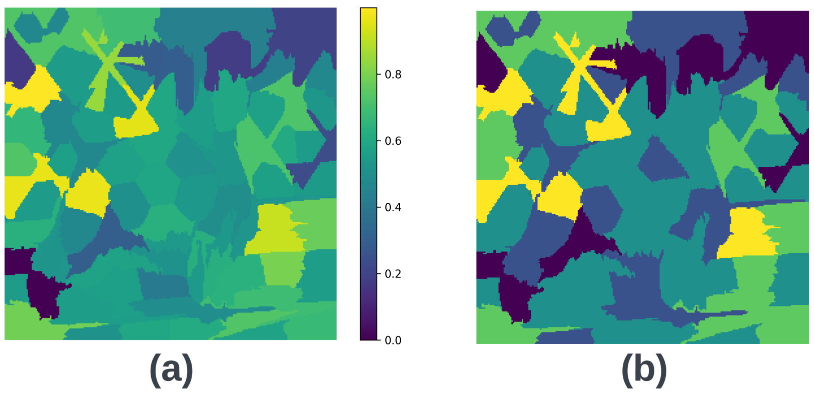


Figure 4.5: Clustering of the fused sensitivity map. **(a)** Fused sensitivity map. **(b)** Fused sensitivity map with superpixels grouped into clusters using k-means clustering.

4.3.5. Clustering

After generating multi-scale superpixels, calculating their sensitivities and fusing them together, we need a way to identify the most relevant regions for inpainting. Simply selecting the top k most sensitive superpixels might overlook groups of superpixels that contribute significantly to the classification as a whole. This is where clustering comes in.

By applying k-means clustering [40] to the sensitivity scores of the fused map, we group together superpixels with similar sensitivity levels. Figure 4.5 visualises the result of this clustering process on the sensitivity map. This allows us to identify clusters of superpixels that collectively influence the model's decision. This is beneficial for several reasons: it captures the inherent structure of important image regions, potentially including regions a rank-based selection might miss. It also ensures that connected regions, rather than isolated superpixels are targeted for inpainting. Finally, we obtain the mask M by clustering the sensitivity regions and selecting the top r most sensitive clusters. This mask is then used for the inpainting process.

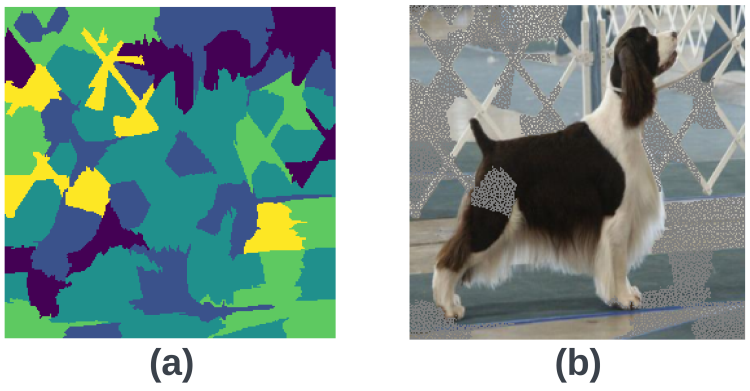


Figure 4.6: Connecting sensitivity to the inpainting mask. **(a)** Sensitivity map. **(b)** Masked image, with the mask (in gray) derived from the sensitivity map and applied to the regions identified as most sensitive.

4.4. Restore ground truth labels via DDPM-based inpainting

Once the most sensitive regions of the image are identified and masked, the MSID uses an inpainting technique to restore the masked areas and remove adversarial perturbations. Figure 4.6 shows an example of an image masked. MSID uses the recently proposed DiffPIR framework [132] to perform inpainting. DiffPIR uses the Half-Quadratic Splitting (HQS) algorithm [32] to address the optimization problem presented in equation 4.2:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma_n^2} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \lambda \mathcal{P}(\mathbf{x}), \tag{4.2}$$

HQS separates the data term and the prior term, allowing for iterative solutions to the resulting subproblems. This decoupling strategy facilitates the integration of a diffusion sampling framework, as Zhang et al. demonstrated in [129]. Equation 4.2 is decoupled into two subproblems, 4.3 and 4.4, through the introduction of an auxiliary variable z. These subproblems are then solved iteratively.

$$z_k = \arg\min_{z} \frac{1}{2(\lambda/\mu)^2} \|z - x_k\|^2 + \mathcal{P}(z), \tag{4.3}$$

$$x_{k-1} = \arg\min_{x} \|y - \mathcal{H}(x)\|^2 + \mu\sigma_n^2 \|x - z_k\|^2, \tag{4.4}$$

Prior Subproblem: This subproblem (see equation 4.3) involves enforcing the prior probability distribution of the image, which is modeled by a denoising diffusion model. The prior term is solved using a Gaussian denoising operator, where the goal is to recover the clean image from the noisy image.

Data Subproblem: This subproblem (see equation 4.4) enforces consistency with the observed data y given the degradation operator $\mathcal{H}(\cdot)$. For inpainting, $\mathcal{H}(\cdot)$ represents a mask that hides the perturbation region, and the data subproblem aims to find an image that matches the unmasked pixels while respecting the prior imposed by the DDPM.

To establish a connection between equation 4.3 and the diffusion process, consider the objective of recovering a noise-free image $\mathbf{z}_k$ from a noisy image $\mathbf{x}_t$ with a noise level $\bar{\sigma}_t$ defined as $\sqrt{\frac{\lambda}{\mu}} = \bar{\sigma}_t$.

Given the noise schedule $\{\beta_t\}$ and the hyperparameter λ, which acts as a guidance scaling parameter, much like in classifier-free diffusion models, the value of $\bar{\sigma}_t$ is known. Equation 4.3 may be interpreted as a proximal operator. Recognizing that the gradient of the negative log-likelihood of the image prior $\mathcal{P}(\mathbf{x})$ is equivalent to the negative score function $-s_\theta(\mathbf{x})$, we can reformulate equation 4.3 as:

$$z_k \approx x_k + \frac{1-\overline{a_t}}{\bar{a}_t} s_\theta(x_k) \tag{4.5}$$

This implies that z_k represents the estimated clean image x_0^t obtained using the "Variance Exploding" Stochastic Differential Equation (SDE) formulation of diffusion models, where $s_\theta(x_k)$ denotes the score function parameterizing the diffusion mode. For clarity, equations 4.3 and 4.4 can be expressed as a three-step process:

$$x_0^{(t)} = \arg\min_z \frac{1}{2\bar{\sigma}_t^2}\|z - x_t\|^2 + P(z), \tag{4.6}$$

$$\hat{x}_0^{(t)} = \arg\min_x \|y - \mathcal{H}(x)\|^2 + \rho_t\|x - x_0^{(t)}\|^2, \tag{4.7}$$

$$x_{t-1} \leftarrow \hat{x}_0^{(t)}, \tag{4.8}$$

where $\rho_t = \lambda\left(\frac{\sigma_n}{\bar{\sigma}_t}\right)^2$. Equation represents the denoising step leveraging the diffusion prior, aiming to find the most probable noise-free image $x_0^{(t)}$ given the noisy input x_t. Subsequently, equation functions as the data term, refining the denoised image $x_0^{(t)}$ by incorporating the observed data y and the forward operator $H(\cdot)$. Finally, equation updates the image estimate for the next iteration of the algorithm.

The authors of DiffPIR observed that the noise term may not provide sufficient perturbation. Therefore, they introduced a hyperparameter ζ to control noise injection. This modification leads to the explicit formulation presented in equation 4.9. The hyperparameter ζ governs the variance of the noise injected at each step. In particular, the sampling strategy becomes deterministic when ζ is set to 0.

$$\sqrt{\bar{\alpha}_{t-1}}\hat{x}_0^{(t)} + \sqrt{1-\bar{\alpha}_{t-1}}(\sqrt{1-\zeta}\hat{\epsilon} + \sqrt{\zeta}\epsilon_t) \tag{4.9}$$

Algorithm 1 provides a comprehensive outline of the DiffPIR algorithm.

Algorithm 1 DiffPIR

```
Require: $s_\theta$, $T$, $y$, $\sigma_n$, $\{\bar{\sigma}_t\}_{t=1}^T$, $\zeta$, $\lambda$

1: Initialize $x_T \sim \mathcal{N}(0, I)$, pre-calculate $\rho_t \triangleq \lambda\frac{\sigma_n^2}{\bar{\sigma}_t^2}$.
2: for $t = T$ to $1$ do
3:     $x_0^{(t)} = \sqrt{\frac{1}{\bar{\alpha}_t}}(x_t + (1-\bar{\alpha}_t)s_\theta(x_t, t))$        ▷ Predict $\hat{z}_0$ with score model as denoiser
4:     $\hat{x}_0^{(t)} = \arg\min_x \|y - H(x)\|^2 + \rho_t\|x - x_0^{(t)}\|^2$        ▷ Solving data proximal subproblem
5:     $\hat{\epsilon} = \sqrt{\frac{1}{1-\bar{\alpha}_t}}(x_t - \sqrt{\bar{\alpha}_t}\hat{x}_0^{(t)})$        ▷ Calculate effective $\hat{\epsilon}(x_t, y)$
6:     $\epsilon_t \sim \mathcal{N}(0, I)$
7:     $x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0^{(t)} + \sqrt{1-\bar{\alpha}_{t-1}}(\sqrt{1-\zeta}\hat{\epsilon} + \sqrt{\zeta}\epsilon_t)$        ▷ Reverse diffusion sampling
8: end for
9: return $x_0$
```

When it comes to inpainting, DiffPIR demonstrates significant advantages over other diffusion-based methods. Its ability to handle arbitrary degradation operators, as opposed to DDRM's [47] linear operators, makes it particularly well-suited for inpainting tasks where the missing regions can be represented by complex masks. Furthermore, DiffPIR is faster compared to Diffusion Posterior Sampling (DPS) [17], especially when dealing with large or complex missing areas. The enhanced speed does not come at the cost of quality, DiffPIR consistently produces high-fidelity inpainted images, even with fewer sampling steps, outperforming the reconstruction accuracy of DPS in such scenarios.

4.4.1. Contextual cues

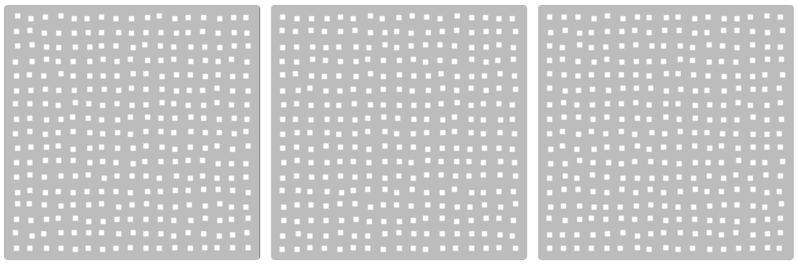


Figure 4.7: Examples of jittered grids for providing contextual cues during inpainting. Each subfigure shows a different realization of the jittered grid applied to the same masked region. The variation in grid point placement helps avoid deterministic patterns and provides more diverse contextual information.

While masking the sensitive regions identified by the clustering previously is a key step for purifying adversarial perturbations, completely removing these areas can obstruct the inpainting process. Inpainting models rely heavily on surrounding context to accurately reconstruct masked pixels [64]. To address this, MSID incorporates contextual cues within the masked regions. we use a strategy of selectively unmasking certain pixels within the masked region. This provides the inpainting model with contextual information, allowing it to better understand the underlying structure and content of the masked area. This, in turn, leads to more accurate and realistic inpainting results. To aid this process, we use a square grid with some randomness (See DDPM-based inpainting part in Figure 4.1). Each point on this grid represents the top-left corner of a square, and we introduce a slight "jitter" to each point's position. This creates a more organic arrangement, subtly shifting each point from its original grid location. Figure 4.7 shows several examples of this jittered grid approach, illustrating the variation in unmasked pixel placement. This approach helps to avoid deterministic patterns and ensures a more realistic distribution of sampling points, ultimately contributing to the effectiveness of our inpainting method.

4.4.2. Removing artifacts

Our AP defense leverages a masking and inpainting approach. To ensure the inpainting model retains sufficient context for accurate reconstruction, we unmask a set of pixels within the masked sensitive regions using a jittered square grid. While this provides contextual cues, it introduces a new challenge: the unmasked pixels themselves become artifacts within the inpainted image. To address this, we introduce a final refinement step using Variance Preservation Sampling (VPS) technique [118]. Figure 4.8 illustrates the effect of VPS on removing the grid artifacts from the inpainted image. VPS operates by iteratively refining the image representation within the diffusion model's latent space, guiding it towards a high probability region that corresponds to clean, artifact-free images. This process effectively eliminates the unwanted unmasked pixels while preserving the integrity of the underlying image structure and the quality achieved in the initial inpainting step.

The core idea is to guide a degraded image towards a clean sample by ensuring it follows the learned distribution of the pre-trained model, while remaining similar to the original degraded input. This is achieved through a two-stage process: ODE inversion for faithfulness and VPS for restoration.

First, given a degraded image y, we approximate invertibility of the diffusion model's ODE sampling process. Specifically, Denoising Diffusion Implicit Models (DDIM) inversion is used to find a corresponding latent representation y_τ:

$$y_\tau = \text{DDIM}^{-1}(y) \tag{4.10}$$

The parameter τ controls the strength of this inversion. While y_τ effectively encodes the degraded image, it typically resides in a low-probability region of the diffusion model's latent space, making direct generation of a high-quality restoration unlikely. Subsequently, VPS refines the latent y_τ by iteratively guiding it towards a nearby high-probability region, representing the distribution of clean images learnt by the diffusion model. This iterative refinement consists of two steps at each timestep $t \in [\tau, 0)$: First, the latent is updated M times using a combination of the gradient of the log-probability density (computed using the pre-trained diffusion model) as follow:

$$y_t^m = y_t^{m-1} + \eta_l \nabla \log p_t \left(y_t^{m-1} \right) + \eta_g \epsilon^m \tag{4.11}$$

such that η_l and η_g are bound by the constraint:

$$\eta_l = \gamma(1 - \bar{\alpha}_t), \quad \eta_g = \sqrt{\gamma(2-\gamma)}\sqrt{1 - \bar{\alpha}_t}. \tag{4.12}$$

where γ is a scalar within the range $0 < \gamma < 1$ that defines the step size, while $\bar{\alpha}_t$ represents the noise schedule from equation 2.1. After the variance preservation step, a DDIM step is applied to further denoise and refine the latent:

$$y_{t-1} = \text{DDIMStep}(y_t^M)$$

This two-step process progressively guides the latent towards the high-probability region associated with clean images. The gradient term $\nabla \log p_t(y_t^{m-1})$ is efficiently computed using the pre-trained diffusion model, specifically by relating it to the predicted noise ϵ_θ:

$$\nabla \log p_t \left(y_t^{m-1} \right) = -\frac{\epsilon_\theta \left(y_t^{m-1}, t \right)}{\sqrt{1 - \bar{\alpha}_t}}$$

By iteratively applying VPS, we remove the artifacts of the degraded image by leveraging the priors captured by the pre-trained diffusion model.

Since our primary goal is to eliminate the artifacts introduced by the masking, we opted for a fast inverse approach in our implementation. Instead of using the DDIM inverse, we used the forward diffusion described in Equation 2.1.

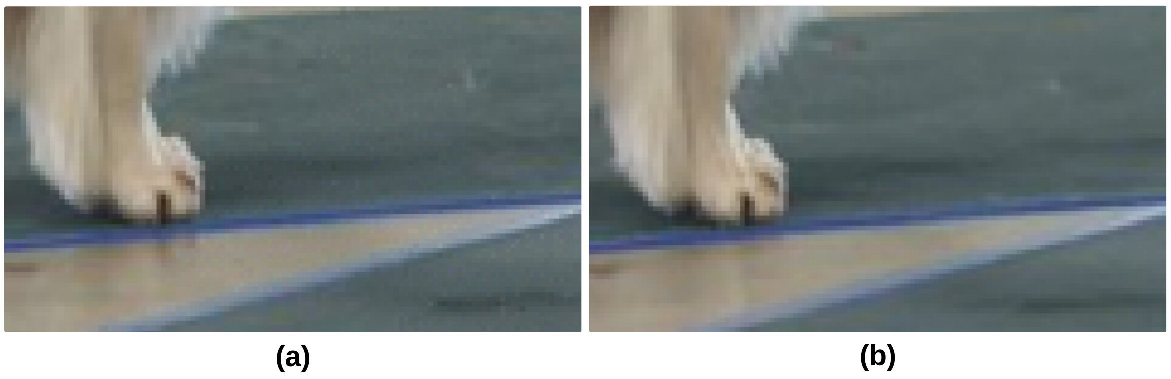


Figure 4.8: Artifact removal using Variance Preservation Sampling (VPS). (a) Inpainted image with grid artifacts. (b) Final refined image after applying VPS. Note the removal of the grid artifacts while preserving the overall image structure and quality.

Bridging the gap between theoretical formulation and practical implementation, Algorithm 2 presents the pseudo-code for the removing artifacts algorithm, showing the mathematical principles described above.

Algorithm 2 Removing artifacts algorithm

```
Require: $y, \tau, M, \eta_l, \eta_g$
Require: A pre-trained diffusion model $\epsilon_\theta$
 1: $y_0 \leftarrow y$ # DDIM inversion, producing the latent $y_\tau$
 2: $y_\tau = \sqrt{\bar{\alpha}_\tau} y_0 + \sqrt{1 - \bar{\alpha}_\tau} \epsilon$
 3: for $t = \tau$ to $1$ do
 4:     $y_0^t \leftarrow y_t$ # Variance Preservation Sampling, no change to $t$
 5:     for $l = 0$ to $L - 1$ do
 6:         $y_{l+1}^t \leftarrow y_l^t - \eta_l \frac{\epsilon_\theta(y_l^t, t)}{\sqrt{1-\bar{\alpha}_t}} + \eta_g \epsilon$
 7:     end for
 8:     $y_t \leftarrow y_L^t$ # DDIM Step, from $t$ to $t - 1$
 9:     $y_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left( \frac{y_t - \sqrt{1-\bar{\alpha}_{t-1}} \epsilon_\theta(y_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \epsilon_\theta(y_t, t)$
10: end for
11: return $y_0$
```

5 Experimental setup

5.1. Threat model

This thesis assumes a threat model where the attacker's primary goal is to induce misclassifications in the target image classifier. We focus on a gray-box scenario, where the attacker has complete knowledge of the classifier's architecture and parameters but no knowledge of the deployed defense mechanism (MSID). This reflects a realistic scenario where attackers may target publicly available or widely-used classifiers without specific knowledge of the defenses employed.

Regarding capabilities, the attacker can craft adversarial examples using various attack methods, including white-box attacks like PGD [65] and AutoAttack [19], as well as black-box attacks like SPSA [104] and color-based attacks like cAdv [9]. The attacker's access to data is limited to the input image and the corresponding classification label. They do not have access to the training data used for either the classifier or the defense.

Due to the computational load of evaluating against adaptive attacks that specifically target the defense mechanism, the majority of our experiments are conducted within gray-box setting. However, to provide a more comprehensive assessment of MSID's robustness, we also include an experiment using the PGD attack augmented with Backward Pass Differentiable Approximation (BPDA) [5] and Expectation Over Transformation (EOT) [6], where the identity function is used for BPDA. This additional experiment provides insights into MSID's performance when facing an attacker with increased knowledge of the defense mechanism, specifically its non-differentiable and stochastic components.

5.2. Experimental setup

We implemented our work entirely in PyTorch [[77], using attacks provided by advertorch [23], the Adversarial Robustness Toolbox [74], and Torchattacks [49] to rigorously test our defense. Experiments were conducted on a dedicated research cluster, specifically on a server with an NVIDIA Tesla V100 GPU, 32GB of RAM, and an AMD EPYC 7402 24-core 2.80 GHz CPU. This robust hardware enabled comprehensive testing of our defense across various scenarios and configurations.

5.3. Datasets

We evaluated our approach using four established datasets widely used in the research community and featured in prior studies [75, 55, 61]. These datasets, chosen for their relevance to our research and their ability to reveal the effectiveness of our defense framework, allow for a fair and comparative assessment of our results, aligning with previous work in the field. To illustrate the characteristics of each dataset, we present a visual overview of representative samples (See Figure 5.1). This allows us to highlight key features and differences between the datasets, providing a clearer understanding of the experimental context.

Figure 5.1: Sample images from the four datasets used in our experiments: From left to right: CIFAR-10, Imagenette, Celeba-HQ, Imagenet

5.3.1. CIFAR-10

CIFAR-10 [50] is a cornerstone dataset in computer vision research, especially for benchmarking image classification models. Containing 60,000 32x32 color images evenly distributed across 10 classes (6,000 images per class), it provides a balanced and manageable dataset well-suited for investigating various facets of image recognition. The standard division into 50,000 training images and 10,000 test images facilitates consistent and comparable evaluation.

Although smaller than many contemporary large-scale datasets, CIFAR-10's manageable size is a significant advantage, enabling rapid experimentation and prototyping. This makes it particularly useful for exploring novel architectures, training methodologies, and defense mechanisms, which is our focus here. Its widespread use as a benchmark offers a strong baseline for comparison, clarifying the performance gains achieved by our proposed approach. In this study, we use CIFAR-10 specifically to evaluate the robustness of our defense mechanism. Table 5.1 presents the main characteristics of the CIFAR-10 dataset.

Table 5.1: Main characteristics of the CIFAR-10 Dataset

Feature	**Description**
Dataset Name	CIFAR-10
Number of Samples	60,000
Number of Classes	10
Image Dimensions	32x32 pixels
Channels	RGB (3 channels)
Data Split	Training (50,000 samples), Testing (10,000 samples)
Annotations	Class labels
Source	Canadian Institute for Advanced Research (CIFAR)

5.3.2. Imagenet

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset [84] has been instrumental in driving progress within computer vision, especially in tasks like image classification, object detection, and localization. Containing over 1.2 million images labeled across 1,000 object categories, ImageNet offers a diverse and extensive benchmark for evaluating deep learning models. Our experimental setup uses ILSVRC 2012 classification subset, comprising 1,281,167 images designated for training, 50,000 for validation, and 100,000 for testing. Using ImageNet allows us to rigorously assess the performance of our proposed method against a well-established and challenging benchmark. Moreover, the widespread use of ImageNet within the research community facilitates direct comparison with state-of-the-art approaches and promotes reproducible research. Table 5.4 presents the main characteristics of the Imagenet dataset.

Table 5.2: Main characteristics of the Imagenet Dataset

Feature	**Description**
Dataset Name	Imagenet
Number of Samples	1,431,167
Number of Classes	1,000
Image Dimensions	varying sizes
Channels	RGB (3 channels)
Data Split	Training (1,281,167 samples), Validation (50,000 samples), Testing (100,000 samples)
Annotations	Class labels
Source	Stanford University

5.3.3. Celeb-HQ Facial Identity Recognition

The CelebFaces Attributes Dataset (CelebA) [62] and its high-resolution counterpart, CelebA-HQ [46], have become benchmark datasets in computer vision research, particularly in areas like face attribute recognition, generative adversarial networks (GANs), and face manipulation detection. CelebA-HQ, derived from CelebA, offers a curated subset of 30,000 high-resolution (1024x1024 pixels) celebrity face images, providing significantly richer detail compared to CelebA's 202,599 images at 178x218 pixels.

Derived from the CelebA-HQ dataset [62, 46], the Celeb-HQ Facial Identity Recognition Dataset, initially introduced with the PyTorch implementation of Latent-HSJA [71], offers a focused benchmark for facial identity classification. While CelebA-HQ provides a large collection of high-resolution celebrity faces with attribute annotations, this curated subset emphasizes identity recognition. Specifically, it features 307 distinct celebrity identities, each represented by at least 15 images, totaling 5,478 high-resolution (1024x1024 pixels) images. The dataset is split into 4,263 images for training and 1,215 for testing. Table 5.3 presents the main characteristics of the Celeb-HQ Facial Identity Recognition dataset.

Table 5.3: Main characteristics of the Celeb-HQ Facial Identity Recognition Dataset

Feature	**Description**
Dataset Name	Celeb-HQ Facial Identity Recognition
Number of Samples	5,478
Number of Classes	307
Image Dimensions	varying sizes
Channels	RGB (3 channels)
Data Split	Training (4,263 samples), Testing (1,215 samples)
Annotations	Class labels
Source	Pohang University of Science and Technology

5.3.4. Imagenette

Imagenette [28] offers a carefully curated subset of the vast ImageNet dataset, purpose-built for rapid prototyping and experimentation in computer vision. While ImageNet boasts an impressive scale of over 1.4 million images across 1000 classes, Imagenette provides a more manageable size, comprising 13,394 images across 10 easily distinguished classes. This smaller, yet representative sample minimizes the subtle inter-class confusion often encountered with larger, more granular datasets. By preserving ImageNet's visual diversity while significantly reducing the computational burden, Imagenette empowers researchers to efficiently explore and validate new ideas before scaling up to the full complexity of ImageNet. Table 5.3 presents the main characteristics of the Imagenette dataset.

Table 5.4: Main characteristics of the Celeb-HQ Facial Identity Recognition Dataset

Feature	Description
Dataset Name	Imagenette
Number of Samples	13,394
Number of Classes	10
Image Dimensions	varying sizes
Channels	RGB (3 channels)
Data Split	Training (9,469 samples), Testing (3,925 samples)
Annotations	Class labels
Source	fast.ai

5.4. Defense and classifier architectures

5.4.1. Defense

Our defense method uses a DDPM-based inpainting method to inpaint potentially corrupted image regions. Our experiments used three pre-trained DDPM models, specifically designed for CIFAR-10, Celeba-HQ and Imagenet. The detailed configurations of the pre-trained DDPM models used in our experiments are provided in the Appendix A.1.

5.4.2. Classifiers

We tested the robustness of our defense method across a diverse range of network architectures, spanning the spectrum from convolutional neural networks (CNNs) and vision transformers (ViTs) to lightweight models. Our evaluation focused on four key architectures: the widely-used ResNet [41] ViT [26], a pure transformer model; the lightweight MobileNetV2 [87], designed for mobile and embedded applications; and WideResNet (WRN) [124], a ResNet variant known for its efficient balance of width and depth, providing rich features with reduced computational costs. This range of models lets us see how well our defense generalizes across various architectural designs. Table 5.5 details which architectures were tested with which datasets, along with their respective Top-1 accuracies, providing a clear picture of the experimental setup.

Table 5.5: Experimental Setup: Architectures, Datasets, and Top-1 Accuracy

Dataset	Architecure	Top-1 Accuracy
CIFAR-10	WideResNet-28-10	94.55
	WideResNet-70-16	94.61
Imagenette	ResNet-50	90.19
	MobileNetv2	98.64
Celeb-HQ Facial Identity Recognition	ResNet-18	89.05
Imagenet	ResNet-50	80.86
	WideResNet-50-2	81.60
	ViT	82.51

5.5. Evaluation metrics

We assessed the effectiveness of the MSID using two key metrics: standard accuracy (on clean data) and robust accuracy (on adversarially perturbed data).

5.5.1. Standard accuracy

Standard accuracy measures the performance of a model on clean, unperturbed data. It represents the model's ability to correctly classify images under normal conditions, free from adversarial attack. In the context of evaluating an adversarial purification defense like MSID, standard accuracy provides a baseline performance indicator. It reveals how the defense affects the model's ability to classify benign inputs. A good defense should maintain high standard accuracy, minimizing any performance degradation on clean data while improving robustness against adversarial attacks. This metric is calculated as the percentage of correctly classified clean images in the test set (see Equation 2.4).

5.5.2. Robust accuracy

Robust accuracy, in contrast, quantifies the model's resilience against adversarial attacks. It measures the classification accuracy on adversarially perturbed images after they have been processed by the defense mechanism. We generate adversarial examples by targeting the model and then measure the model's performance (with the defense applied) on these perturbed images (gray-box scenario). The percentage of these correctly classified adversarial examples constitutes the robust accuracy (see Equation 2.5). Following the approach of Nie et al. [75], we calculate robust accuracy using a randomly sampled subset of 512 images from the test set to manage the computational load of evaluating against a comprehensive range of attacks. This subset provides a representative sample for assessing the defense's effectiveness.

5.6. Adversarial attacks

In order to evaluate the effectiveness of our defense, we tested it to a range of diverse adversarial attacks, including the AutoAttack benchmark [19]. This section details the specific attacks used in our evaluation, ranging from spatially transformed adversarial examples and projected gradient descent variations to score-based black-box attacks and colorization attack. We also incorporate attacks based on backward pass differentiable approximation with expectation over transformation (BPDA+EOT). These methods, described in Sections 5.6.1 through 5.6.5, cover a comprehensive set of adversarial techniques, allowing us to assess the robustness of our defense against various attack strategies. As a starting point, we evaluate our defense under preprocessor-blind scenario [110, 123]. In this attack scenario, sometimes referred to as a gray-box attack, the attacker has full access to the classifier f_ϕ but no access to the purification model g_θ. We use the torchattacks library [49] for AutoAttack, PGD, and SPSA, while BPDA+EOT attacks were implemented using advertorch [23]. The colorization attack used the implementation provided by the authors of ”Unrestricted Adversarial Examples via Semantic Manipulation” [9], and the StAdv (spatially transformed adversarial examples) attack employed its original implementation [117]. Understanding the specifics of these attacks is essential for interpreting the performance evaluations presented in later sections.

5.6.1. AutoAttack

An important aspect of evaluating any defense mechanism is its performance against strong, adaptive attacks. For this purpose, we use the AutoAttack benchmark [19], a powerful and widely-used framework for evaluating adversarial robustness. AutoAttack exists in two versions: (i) the STANDARD version (APGD-CE, APGD-T, FAB-T, and Square) primarily used for evaluating deterministic defenses, and (ii) the RAND version (APGD-CE and APGD-DLR) designed for evaluating stochastic defenses. Due to the stochastic nature of our method, we choose the RAND version of AutoAttack. We tested our defense against this benchmark using both the l_∞ and l_2 norms, representing common threat models in adversarial robustness evaluation. These norms constrain the magnitude of the adversarial perturbation, with l_∞ limiting the maximum change in any individual pixel and l_2 limiting the overall magnitude of the perturbation vector. Evaluating against both norms offers a more comprehensive assessment of our defense's effectiveness under different attack constraints.

AutoAttack employs two distinct attacks, each designed to exploit different vulnerabilities in defenses. We will now enumerate and briefly describe each of these attacks. They are:

Auto Projected Gradient Descent - Cross-Entropy (APGD-CE): This white-box attack iteratively minimizes the cross-entropy loss of the model's predictions with respect to the true label [19]. Unlike standard PGD, APGD-CE dynamically adjusts its step size throughout the attack, improving its ability to find strong adversarial examples. It also incorporates a momentum term to accelerate convergence and escape local minima. The perturbation is constrained within the specified ℓ_p-norm ball (e.g., ℓ_∞ or ℓ_2) by projecting the updated adversarial example onto the ball after each iteration.

Auto Projected Gradient Descent - Difference of Logits Ratio (APGD-DLR): APGD-DLR is a variant of the Auto-PGD (APGD) attack that uses the Difference of Logits Ratio (DLR) loss function. The standard cross-entropy (CE) loss used in adversarial attacks suffers from issues related to logit scaling: rescaling the logits can artificially inflate robustness estimates. The DLR loss addresses this by being both shift and scale-invariant, meaning that arbitrary rescaling of the model's output logits does not affect the loss or its gradient. This makes DLR more robust to gradient masking, a defense mechanism where the model appears robust due to vanishing or misleading gradients.

5.6.2. Projected Gradient Descent

Projected Gradient Descent (PGD) [65] is a widely used white-box adversarial attack known for its effectiveness in generating strong adversarial examples [110, 123, 10]. It iteratively perturbs an input image in the direction of the gradient of the loss function with respect to the input, while constraining the perturbation within a specified L_p-norm ball. PGD effectively explores the boundaries of the adversarial space within the L_p-norm constraint, making it a strong benchmark for evaluating the robustness of defense methods. Due to its strength and widespread adoption, PGD serves as a cornerstone in adversarial robustness evaluation [18].

5.6.3. Spatially Transformed Adversarial Examples

Traditional adversarial attacks often focus on perturbing pixel values directly. However, these l_p-norm constrained perturbations can sometimes introduce noticeable artifacts. To evaluate the robustness of our defense against more perceptually realistic attacks, we use Spatially Transformed Adversarial Examples (StAdv) [117]. StAdv generates adversarial examples by applying smooth spatial deformations to the input image rather than directly manipulating pixel intensities. This approach optimizes a transformation flow field that displaces pixels to minimize a combination of the adversarial loss and a regularization term promoting smooth deformations. In our experiments, we use StAdv with a non-l_p constraint, setting the regularization parameter ϵ to 0.05. This parameter controls the smoothness of the spatial transformation, with smaller values leading to smoother deformations. Because StAdv operates in the spatial domain, it can generate adversarial examples that are perceptually similar to the original image while potentially bypassing defenses designed to mitigate l_p-norm bounded attacks. Therefore, incorporating StAdv in our evaluation provides a more comprehensive assessment of our defense's robustness against a wider range of adversarial strategies.

5.6.4. Simultaneous Perturbation Stochastic Approximation

Simultaneous Perturbation Stochastic Approximation (SPSA) is a black-box adversarial attack that efficiently generates adversarial examples by estimating the gradient of a target model's loss function with respect to the input image. Instead of requiring explicit gradient computations, as in white-box attacks, SPSA utilizes a finite-difference approximation of the gradient. This makes SPSA suitable for attacking scenbarios where gradient information is unavailable. In each iteration, SPSA perturbs the input image with a randomly generated vector and evaluates the model's loss on both the perturbed and original images. This difference in loss, combined with the perturbation vector, provides an estimate of the gradient, which is then used to update the adversarial example in a direction that increases the model's loss, thus misclassifying the input. The stochastic nature of the perturbation vector aids in escaping local optima and finding more robust adversarial examples.

Therefore, testing a purification defense against SPSA provides a much more rigorous evaluation of its robustness. In our experiments, we used 1,280 queries to ensure a sufficiently powerful attack.

5.6.5. Colorization Attack

The colorization attack (cAdv) [9] creates adversarial examples that leverages color manipulations of images rather than the typical small, imperceptible perturbations constrained by L_p norms. Instead of focusing on minimizing the difference between the original and adversarial example in terms of pixel values, this attack exploits how deep learning models interpret higher-level features like colors.

cAdv manipulates the colors of an image using a pre-trained colorization model. By altering the colorization process, cAdv introduces smooth, consistent, and often large color changes that, while potentially significantly different from the original image in terms of L_p norm, appear photorealistic to humans. This is achieved by carefully selecting which areas of the image to recolor based on color ambiguity, preserving colors in regions like roads where the color is generally fixed, while allowing more flexibility in regions where color variation is expected, such as an umbrella.

The key idea behind these attack is that it exploits the sensitivity of DNNs to colors, demonstrating that even large, semantically-driven changes can fool these models while remaining relatively unnoticeable to humans.

5.6.6. Overcoming non-differentiability and stochasticity with BPDA+EOT

Adaptive defenses, including many existing adversarial purification methods, often involve iterative optimization or sampling at test time [55]. These processes under white-box scenario can introduce non-differentiable operations and stochastic behavior, making them incompatible with strong adaptive attacks like AutoAttack, which requires full gradients [75]. While our defense was previously evaluated under gray-box scenario, we adopt the BPDA+EOT attack [20] for a fair comparison with existing defenses. BPDA addresses the non-differentiability by approximating these operations with differentiable functions during the backward pass of gradient computation. We use the identity function for this approximation, effectively treating the non-differentiable operation as if it did not affect the gradient. Meanwhile, EOT (Expectation over Transformation) [5] handles the stochasticity present in some defenses, such as adding noise to inputs, using stochastic optimization, or relying on random augmentations—by averaging the predictions and gradients over multiple runs with the same input. This effectively computes the expected behavior of the defense. In our experiments, we used PGD+BPDA+EOT with 40 PGD steps and 20 EOT iterations.

5.7. Implementation details

All experiments were implemented in PyTorch [77] and conducted on an NVIDIA V100 GPU.

5.7.1. Multi-Scale superpixel segmentation hyperparameters

To capture image details at multiple scales, we used multi-scale superpixel segmentation based on SLIC [1]. The specific scales and corresponding weights are provided in Table 5.6.

For the SLIC algorithm, we set the compactness parameter to 10 and used the default values for all other parameters as provided in the scikit-image library [108].

Table 5.6: Scales and weights for multi-scale superpixel segmentation

Dataset	**Superpixels per Scale**	$w1$	$w2$	$w3$
CIFAR-10	5, 10, 15	0.2	0.3	0.5
ImageNet	100, 200, 300	0.2	0.3	0.5
Celeb-HQ Facial Identity Recognition	100, 200, 300	0.2	0.3	0.5
Imagenette	100, 200, 300	0.2	0.3	0.5

5.7.2. DIFFPIR hyperparameters

DIFFPIR was configured with the hyperparameters outlined in Table 5.7, which were selected via empirical optimization.

Table 5.7: Hyperparameters for DIFFPIR

Hyperparameter	Description	Value
T	Timesteps	20
u	Resampling iterations	20
λ	Conditioning guidance strength	10
ζ	Noise level	0.5

5.7.3. Artifact removal hyperparameters

Artifact Removal introduces additional hyperparameters, which were optimized empirically and are shown in Table 5.8

Table 5.8: Hyperparameter values for artifact removal

Hyperparameter	Description	Value
τ	Inverse strength	10
γ	Step size	0.05
M	Refinement Iterations (per timestep)	1

6 Results and discussion

This chapter presents and discusses the experimental results of the proposed defense method. Section 6.1.1 benchmarks its performance against state-of-the-art defenses on several datasets using standard and robust accuracy metrics under AutoAttack and PGD. The chapter then explores the method's robustness against various attacks: unseen attacks (6.1.2), score-based black-box attacks (6.1.3), color-based attacks (6.1.4), and strong adaptive attacks (6.1.5). Further analysis investigates the impact of contextual cues on adversarial robustness and the contribution of multi-scale information (6.2 and 6.3, respectively). Finally, the chapter evaluates different imputation strategies for creating robust occlusion sensitivity maps (**??**) and various techniques for generating binary masks (**??**), assessing their contributions to the defense's overall effectiveness.

6.1. Experimental results

6.1.1. Comparison with the state-of-the-art

This section presents the experimental results of our defense, comparing the performance of the proposed method with state-of-the-art techniques. Tables below summarize the standard and robust accuracy achieved by our method and existing approaches under AutoAttack and PGD with different perturbation norms l_∞ and l_2. The experiments were conducted across diverse datasets, including CIFAR-10, Celeb-HQ, Imagenette, and ImageNet, using different classifier architectures such as WideResNet, ResNet, MobileNet, and ViT.

Our experiments on CIFAR-10 consistently demonstrate MSID's superior robust accuracy against both AutoAttack (Tables 6.1a and 6.1b) and PGD (Tables 6.2a and 6.2b). Notably, against the highly effective AutoAttack l_∞ (Table 6.1a), MSID achieves robust accuracy of 87.89% and 85.54% with WideResNet-28-10 and WideResNet-70-16, respectively. This outperformance compared to the best baseline methods (e.g., Lin et al. [61] at 82.76% and 80.12%) highlights the effectiveness of MSID's targeted purification approach.

The key reason for this advantage lies in MSID's ability to identify and remove adversarial perturbations without significantly altering benign image features. Unlike global purification methods that might over-smooth the image, MSID focuses its inpainting efforts on the sensitive regions identified by the multi-scale occlusion sensitivity maps. This targeted approach, coupled with the powerful generative capabilities of the DDPM, allows for a more precise restoration, effectively neutralising adversarial noise while preserving benign image features.

Furthermore, the multi-scale analysis contributes significantly to the robustness of the MSID. By analysing the image at different levels of granularity, MSID can capture adversarial perturbations affecting both fine details and larger structural components. The subsequent fusion of these multi-scale sensitivity maps ensures a comprehensive understanding of the perturbed regions, leading to more effective inpainting.

Table 6.1: Standard and robust accuracy against AutoAttack on CIFAR-10 with WideResNet classifier models.

(a) AutoAttack l_∞ ($\epsilon = 8/255$)

	Type	Method	Extra Data	Standard Acc.	Robust Acc.
WideResNet-28-10	AT	Zhang et al. [128]	✓	85.36	59.96
		Wu et al. [115]	✓	88.25	60.04
		Gowal et al. [38]	✓	89.48	62.70
		Cui et al. [21]	×	**92.16**	67.73
	AP	Yoon et al. [123]	×	85.66	68.42
		Lee & Kim [55]	×	90.16	79.11
		Nie et al. [75]	×	89.02	78.21
		Lin et al. [61]	×	90.62	82.76
		Ours	×	90.23	**87.89**
WideResNet-70-16	AT	Rebuffi et al. [82]	×	88.54	64.25
		Gowal et al. [38]	✓	91.10	65.87
		Rebuffi et al. [82]	✓	92.23	66.58
		Wang et al. [112]	×	**93.25**	70.69
	AP	Yoon et al. [123]	×	86.76	60.86
		Nie et al. [75]	×	90.43	74.83
		Lee & Kim [55]	×	90.53	76.35
		Lin et al. [61]	×	91.99	80.12
		Ours	×	90.86	**85.54**

(b) AutoAttack l_2 ($\epsilon = 0.5$)

	Type	Method	Extra Data	Standard Acc.	Robust Acc.
WideResNet-28-10	AT	Augustin et al. [7]	✓	92.23	77.93
		Rebuffi et al. [82]	×	91.79	78.32
		Wang et al. [112]	×	**95.16**	83.68
	AP	Yoon et al. [123]	×	85.66	83.51
		Lin et al. [61]	×	90.62	85.77
		Nie et al. [75]	×	91.4	86.43
		Lee & Kim [55]	×	90.16	87.29
		Ours	×	90.23	**88.28**
WideResNet-70-16	AT	Gowal et al. [38]	×	90.90	74.03
		Rebuffi et al. [82]	✓	**95.74**	81.44
	AP	Nie et al. [75]	×	92.68	85.70
		Lin et al. [61]	×	91.99	86.78
		Lee & Kim [55]	×	90.53	87.39
		Ours	×	90.23	**88.86**

Similar trends are observed under AutoAttack l_2 (Table 6.1b) and PGD attacks (Tables 6.2a and 6.2b), reinforcing MSID's ability to defend against diverse attack strategies. For instance, against PGD l_∞ with WideResNet-28-10 (6.2a), MSID achieves 86.52% robust accuracy, outperforming Lee & Kim [55] (84.03%). This advantage persists even with larger models like WideResNet-70-16, demonstrating the scalability of our defense. It is important to note that while achieving state-of-the-art robust accuracy, MSID maintains a competitive standard accuracy (e.g., 90.23% and 90.86% with WideResNet-28-10 and WideResNet-70-16). This shows that our targeted purification avoids the common trade-off between robustness and standard performance often encountered in adversarial training methods.

Table 6.2: Standard and robust accuracy against PGD on CIFAR-10 with WideResNet classifier models.

(a) PGD l_∞ ($\epsilon = 8/255$)

	Type	Method	Extra Data	Standard Acc.	Robust Acc.
WideResNet-28-10	AT	Pang et al. [76]	✓	88.62	64.95
		Gowal et al. [38]	✓	88.54	65.93
		Gowal et al. [37]	✓	87.51	66.01
	AP	Yoon et al. [123]	×	85.66	79.28
		Nie et al. [75]	×	91.41	80.78
		Wang et al. [110]	×	**93.50**	81.05
		Lee & Kim [55]	×	90.16	84.03
		Ours	×	90.23	**86.52**
WideResNet-70-16	AT	Gowal et al. [38]	✓	91.10	68.66
		Gowal et al. [37]	✓	88.75	69.03
		Rebuffi et al. [82]	×	**92.22**	69.97
	AP	Yoon et al. [123]	×	86.76	74.58
		Nie et al. [75]	×	92.15	79.15
		Wang et al. [110]	×	**93.50**	83.33
		Lee & Kim [55]	×	90.53	83.92
		Ours	×	90.86	**85.54**

(b) PGD l_2 ($\epsilon = 0.5$)

	Type	Method	Extra Data	Standard Acc.	Robust Acc.
WideResNet-28-10	AT	Pang et al. [89]	✓	90.93	83.75
		Rebuffi et al. [82]	×	91.79	85.05
		Augustin et al. [7]	✓	**93.96**	86.14
	AP	Yoon et al. [123]	×	85.66	80.13
		Nie et al. [75]	×	91.41	83.11
		Wang et al. [110]	×	**93.50**	85.76
		Lee & Kim [55]	×	90.16	87.09
		Ours	×	90.23	**89.25**
WideResNet-70-16	AT	Rebuffi et al. [82]	×	92.41	86.24
		Gowal et al. [38]	✓	94.74	88.18
		Rebuffi et al. [82]	×	**95.74**	89.62
	AP	Yoon et al. [123]	×	86.76	78.54
		Nie et al. [75]	×	92.15	82.93
		Wang et al. [110]	×	**93.50**	84.16
		Lee & Kim [55]	×	90.53	86.04
		Ours	×	90.86	**88.08**

The robustness of MSID extends beyond CIFAR-10 to more complex datasets like Celeb-HQ (Tables 6.3a and 6.3b). On the challenging facial identity recognition task on Celeb-HQ, MSID achieves the highest robust accuracy against both AutoAttack and PGD attacks with a ResNet-18 classifier. Against the AutoAttack, MSID achieves a robust accuracy of 57.95% (l_∞, Table 6.3a) and 70.46% (l_2, Table 6.3b). These results represent a clear advantage over existing AP methods on this dataset. For instance, against AutoAttack l_∞, MSID outperforms Yoon et al. [123] (45.90%), Nie et al. [75] (50.10%), Wang et al. [112] (51.66%), and Lee & Kim [55] (55.73%). The margins of improvement are considerable, highlighting the effectiveness of MSID's targeted approach in preserving facial identity under attack.

Table 6.3: Standard and robust accuracy against AutoAttack on Celeb-HQ Facial Identity Recognition with Resnet-18 classifier.

(a) AutoAttack l_∞ ($\epsilon = 4/255$)

Method	**Standard Acc.**	**Robust Acc.**
Yoon et al. [123]	70.05	45.90
Nie et al. [75]	78.79	50.10
Wang et al. [110]	**79.14**	51.66
Lee & Kim [55]	75.08	55.73
Ours	77.23	**57.95**

(b) AutoAttack l_2 ($\epsilon = 0.5$)

Method	**Standard Acc.**	**Robust Acc.**
Yoon et al. [123]	70.05	54.27
Nie et al. [75]	78.79	60.95
Lee & Kim [55]	75.08	64.52
Wang et al. [110]	**79.14**	65.37
Ours	77.23	**70.46**

Similar trends are observed against PGD attacks. MSID achieves 54.02% robust accuracy against PGD l_∞ (Table 6.4a) and an impressive 75.60% against PGD l_2 (Table 6.4b). Again, these figures outperform the baseline methods.

Table 6.4: Standard and robust accuracy against PGD on Celeb-HQ Facial Identity Recognition with Resnet-18 classifier.

(a) PGD l_∞ ($\epsilon = 8/255$)

Method	**Standard Acc.**	**Robust Acc.**
Yoon et al. [123]	70.05	39.52
Nie et al. [75]	78.79	43.18
Lee & Kim [55]	75.08	47.23
Wang et al. [110]	**79.14**	50.73
Ours	77.23	**54.02**

(b) PGD l_2 ($\epsilon = 0.5$)

Method	**Standard Acc.**	**Robust Acc.**
Yoon et al. [123]	70.05	62.28
Nie et al. [75]	78.79	65.19
Lee & Kim [55]	75.08	67.85
Wang et al. [110]	**79.14**	71.94
Ours	77.23	**75.60**

When comparing MSID's performance to baselines on Imagenette, the improvements are significant. MSID consistently achieves state-of-the-art robust accuracies against both AutoAttack and PGD l∞ attacks with both ResNet-50 and MobileNetV2 architectures (see Tables 6.5a and 6.5b). For instance, the robust accuracy gains over Yoon et al. [123] are substantial, indicating that MSID's approach to adversarial purification is more effective in preserving the integrity of the image while removing malicious perturbations on this dataset. The comparison to Nie et al. [75], another diffusion-based purification method, highlights the advantages of MSID's targeted approach compared to potentially global purification strategies. The gains over Wang et al. [112] further solidify MSID's position as a leading adversarial defense method for this dataset.

Table 6.5: Standard and robust accuracy against Autoattack and PGD on Imagenette.

(a) PGD l_∞ ($\epsilon = 4/255$)

	Method	Standard Acc.	Robust Acc.
Resnet-50	Yoon et al. [123]	78.11	70.75
	Nie et al. [75]	82.06	74.32
	Wang et al. [110]	85.41	76.28
	Lee &	**85.44**	78.1
	Ours	84.08	**81.01**
MobileNetV2	Yoon et al. [123]	88.56	80.64
	Nie et al. [75]	92.03	84.96
	Lee &	94.12	87.22
	Wang et al. [110]	**94.73**	88.63
	Ours	93.31	**90.19**

(b) Autoattack l_∞ ($\epsilon = 4/255$)

	Method	Standard Acc.	Robust Acc.
Resnet-50	Yoon et al. [123]	78.11	68.1
	Nie et al. [75]	82.06	71.02
	Wang et al. [110]	85.41	73.33
	Lee &	**87.44**	74.17
	Ours	84.08	**77.34**
MobileNetV2	Yoon et al. [123]	88.56	78.11
	Nie et al. [75]	92.03	83.55
	Lee &	94.12	85.91
	Wang et al. [110]	**94.73**	86.13
	Ours	93.31	**87.33**

The ImageNet dataset presents a challenge for adversarial defenses due to its high dimensionality and complex image variations. Our results on ImageNet (Tables 6.6b and 6.6a) demonstrate that MSID achieves improvements in robust accuracy against both PGD and AutoAttack l_∞ attacks with various architectures (ResNet-50, WRN, and ViT). Particularly, against AutoAttack l_∞ (Table 6.6b), MSID achieves 74.80% robust accuracy with ResNet-50, an increase compared to Nie et al. [75] (66.23%), while also achieving a higher standard accuracy (76.39% vs. 67.79%).

The improvement on ImageNet is a key highlight of MSID. The ability to effectively defend against adversarial attacks on such a complex dataset underscores the robustness and scalability of our approach. The targeted nature of MSID proves particularly important here, preventing the over-smoothing that can severely degrade the performance of global purification methods on high-resolution images.

While the robust accuracy achieved with the ViT architecture on ImageNet is slightly lower compared to the ResNet-based models, it still performs competitively (63.08% against AutoAttack l_∞, Table 6.6b). Nevertheless, the overall performance on ImageNet solidifies MSID's position as a highly effective adversarial defense.

Table 6.6: Standard and robust accuracy against PGD and AutoAttack on ImageNet.

(a) PGD l_∞ ($\epsilon = 4/255$) with ResNet-50 classifier

Type	Method	Standard Acc.	Robust Acc.
AT	Wong et al. [113]	53.83	28.04
	Salman et al. [85]	63.86	39.11
AP	Nie et al. [75]	71.48	50.59
	Lee & Kim [55]	70.74	54.73
	Wang et al. [110]	70.17	67.06
	Ours	**76.39**	**74.60**

(b) AutoAttack l_∞ ($\epsilon = 4/255$)

	Type	Method	Standard Acc.	Robust Acc.
ResNet-50	AT	Wong et al. [113]	55.62	26.95
		Bai et al. [8]	67.30	35.51
		Salman et al. [85]	64.02	37.89
	AP	Nie et al. [75]	67.79	66.23
		Ours	76.39	**74.80**
WRN-50-2	AT	Salman et al. [85]	68.46	39.25
	AP	Nie et al.[75]	71.16	64.21
		Ours	**78.60**	**75.39**
ViT	AT	Bai et al.[8]	66.50	35.50
	AP	Nie et al.[75]	73.63	52.33
		Ours	**75.11**	**63.08**

In summary, the consistent outperformance of MSID across diverse datasets, attack types, and model architectures arises from its core design principles:

Targeted Perturbation Removal: Focusing inpainting efforts only on identified perturbed regions prevents the degradation of benign image features and maintains standard accuracy.

Multi-Scale Occlusion Sensitivity Mapping: Captures adversarial perturbations affecting both fine details and broader structures, leading to more accurate identification of problematic areas.

DDPM Inpainting: Leverages the generative capabilities of diffusion models to realistically restore perturbed regions, effectively neutralising adversarial noise.

Contextual Awareness: The incorporation of contextual cues during inpainting ensures more coherent and accurate reconstructions.

6.1.2. Defense against unseen attacks

A well-known limitation of AT is its tendency to overfit to the specific attacks used during training, resulting in poor generalization to unseen attacks. Table 6.7 clearly illustrates this issue. Standard AT methods, trained against a specific attack (e.g., PGD with l_∞ norm), perform poorly when evaluated against different attacks like PGD with l_2 norm and StAdv. For instance, the model ”AT with l_∞” [53] achieves a robust accuracy of 49.0% against AutoAttack l_∞ but drastically drops to 19.2% and 4.8% against AutoAttack l_2 and StAdv.

In contrast, our method, being an AP technique, demonstrates remarkable robustness across all three unseen attacks. As shown in Table 6.7, MSID achieves robust accuracies of 85.5%, 84.57%, and 86.5% against AutoAttack l_∞, l_2, and StAdv. This outperformance compared to AT methods highlights the inherent generalization capability of purification-based defenses. MSID effectively cleanses the adversarial perturbations regardless of the specific attack method used to generate them.

Compared to other AP defenses like DiffPure [75] and AToP [61], MSID also demonstrates superior performance against these attacks, especially on StAdv. The improvements of 45.5%, 50.67%, and 36.9% over the baseline AT methods on l_∞, l_2, and StAdv, underscore the effectiveness of MSID’s targeted inpainting approach in handling diverse perturbation patterns. This strong generalization ability is a precious advantage for deploying robust deep learning models in real-world scenarios where the nature of potential adversarial attacks might be unknown.

Table 6.7: Standard accuracy and robust accuracy against AutoAttack l_∞ $(\epsilon = 8/255)$, l_2 $(\epsilon = 1)$ and StAdv non-l_p $(\epsilon = 0.05)$ threat models on CIFAR-10 with ResNet-50 model.

Method	**Standard Acc.**	**AA l_∞**	**AA l_2**	**StAdv**
Standard Training	94.8	0.0	0.0	0.0
AT with l_∞ [53]	86.8	<u>49.0</u>	19.2	4.8
AT with l_2 [53]	85.0	39.5	<u>47.8</u>	7.8
AT with StAdv [53]	86.2	0.1	0.2	<u>53.9</u>
AT with all [53]	84.0	<u>25.7</u>	<u>30.5</u>	<u>40.0</u>
PAT-self [53]	82.4	30.2	34.9	46.4
Adv. CRAIG [25]	83.2	40.0	33.9	49.6
DiffPure [75]	88.2	79.57	81.29	68.73
AToP [61]	89.1	83.42	82.44	70.59
Ours	88.7	**85.5**	**84.57**	**86.5**

6.1.3. Defense against score-based black-box attack

Even without direct access to a model or its gradients, attackers can effectively estimate gradients using a large number of samples. One such method, SPSA [104], approximates gradients through a finite-difference approach, sampling points near an input. This gradient estimation allows attackers to craft adversarial perturbations even when model internals are unavailable. To evaluate the robustness of our defense against such attacks, we tested it against SPSA with an l_∞ constraint $(\epsilon = 8/255)$ on CIFAR-10 using WideRestNet-28-10. As Table 6.8 shows, our method achieves state-of-the-art performance, reaching 86.52% robust accuracy—a mere 0.92% below Wang's defense [110]. This indicates that MSID's purification process is effective even when attackers can only estimate gradients through sampling, highlighting its practical relevance in scenarios with limited model access.

Table 6.8: Standard accuracy and robust accuracy against SPSA l_∞ $(\epsilon = 8/255)$ on CIFAR-10 with WideRestNet-28-10.

Method	**Standard Acc.**	**Robust Acc.**
Yoon et al. [123]	86.14	80.80
Wang et al.[110]	**93.50**	**87.44**
Ours	90.23	86.52

6.1.4. Defense against color-based adversarial attack

cADV [9] generates adversarial examples by manipulating the color of an image. It leverages a pre-trained colorization model, guiding it to produce colorizations that misclassify the image while maintaining a benign appearance. This differs from other attacks that typically add small, imperceptible perturbations constrained by L_p norms. cADV, however, introduces large, smooth, and semantically consistent color changes. Instead of minimizing L_p distance, cADV searches the color space for adversarial examples, exploiting the colorization model's inherent understanding of clean color relationships and boundaries. Our method demonstrates significant performance against the cAdv attack on both, CIFAR-10 and Imagenet datasets, as shown in Table 6.9.

Table 6.9: Robust Accuracy of MSID Against cAdv Attack.

Dataset	Model	Robust Acc.
CIFAR-10	Resnet-50	75.19
	WRN-28-10	74.15
	WRN-70-16	72.46
Imagenet	Resnet-50	64.84

6.1.5. Defense against strong adaptive attacks

It is important to note that for AP methods that involve optimization loops or non-differentiable operations, the BPDA attack is widely recognized as the most effective attack [103]. When considering stochastic defense methods, the BPDA+EOT attack has become the standard benchmark for evaluating the latest advancements in adversarial purification [123, 42, 55]. Strong adaptive attacks, designed specifically to bypass a particular defense mechanism, represent the ultimate test of robustness. Table 6.10 presents MSID's performance against the strong adaptive attack BPDA+EOT, a standard benchmark for evaluating stochastic defenses. Our method achieves a robust accuracy of 74.21% against BPDA 40+EOT on CIFAR-10 with WideResNet-28-10. While not the highest robustness, this performance is competitive with state-of-the-art methods like Wang et al. [112] (79.83%) and Yoon et al. [123] (70.01%), indicating a good level of robustness against sophisticated attacks specifically crafted to circumvent purification defenses.

Table 6.10: Standard and robust accuracy against BPDA+EOT on CIFAR-10 with WideResNet-28-10.

Method	Standard Acc.	Robust Acc.
BPDA 50+EOT Hill et al. [42]	84.12	54.9
BPDA 40+EOT Yoon et al. [123]	86.14	70.01
BPDA 40+EOT Wang et al. [110]	**93.50**	**79.83**
BPDA 40+EOT Ours	90.23	74.21

6.2. Impact of contextual cues on robustness to adversarial attacks

Table 6.11: Impact of Contextual Cues on Robustness.

Attack	Dataset	Attack parameters	Model	Robust Acc.
AutoAttack	CIFAR-10	$l_\infty(\epsilon = 8/255)$	WRN-28-10	46.67
	Imagenet	$l_\infty(\epsilon = 8/255)$	Resnet-50	57.96
SPSA	CIFAR-10	$l_\infty(\epsilon = 8/255)$	WRN-28-10	50.39
PGD	CIFAR-10	$l_\infty(\epsilon = 8/255)$	WRN-28-10	51.95
	Imagenet	$l_\infty(\epsilon = 8/255)$	Resnet-50	64.84
StAdv	CIFAR-10	non-$l_p(\epsilon = 0.05)$	Resnet-50	52.92

To understand the importance of the contextual cues incorporated into our inpainting process, we conducted an ablation study where these cues were removed. The results, presented in Table 6.11, unequivocally demonstrate a significant decrease in robust accuracy across all tested attack types and datasets when contextual cues are absent. For instance, against AutoAttack l_∞ on CIFAR-10, the robust accuracy drops from our reported results in Table 6.1a to 46.67%. Similarly, on ImageNet, the robust accuracy against the same attack falls to 57.96%. This drop underscores the important role of providing the DDPM with surrounding information to enable accurate and semantically coherent inpainting, ultimately contributing to the defense's overall effectiveness. Without these cues, the inpainting process is less informed, potentially restoring the masked regions in a way that does not reflect the original image.

6.3. Analysing the contribution of multi-scale information in occlusion sensitivity mapping

The number of scales used in our defense directly influences its robust accuracy. We assessed the impact of varying the number of scales and observed a clear trend (see Figure 6.1). Robust accuracy increased sharply with each additional scale up to three scales. However, beyond three scales, the gains in robustness plateaued. Therefore, to balance the computational overhead introduced by incorporating more scales against the resulting improvement in robust accuracy, we determined that three scales provide the optimal trade-off. While adding scales offer negligible robustness, they require a greater computational load, making three scales the most efficient and effective choice for our defense. This multi-scale approach achieves greater robustness than a single-scale alternative by capturing both finer image details and adversarial perturbations This evaluation was carried out on CIFAR-10 using PGD l_∞ attacks with $\epsilon = 8/255$.

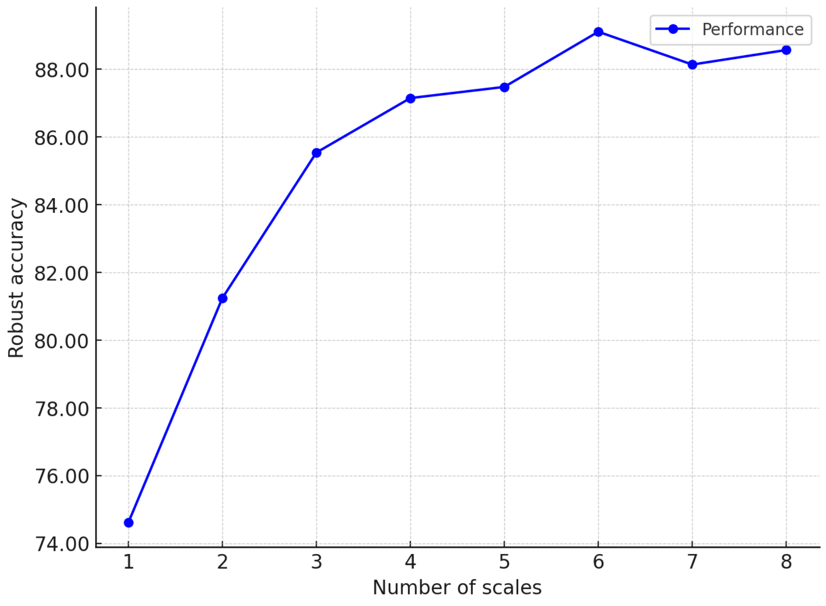


Figure 6.1: Impact of single vs multi-scale sensitivity maps on robust accuracy.

6.4. Evaluation of binary mask generation techniques

This section addresses the research question: "What methods can be used to generate a binary mask from the fused sensitivity map that accurately isolates regions requiring inpainting for effective adversarial purification in MSID?" As described previously, the goal is to identify the optimal strategy for converting the continuous sensitivity map into a binary mask, effectively guiding the inpainting process towards the removal of adversarial perturbations. Two primary methods, thresholding and clustering, were evaluated and compared for their effectiveness in achieving this objective using PGD attacks with $\epsilon = 8/255$.

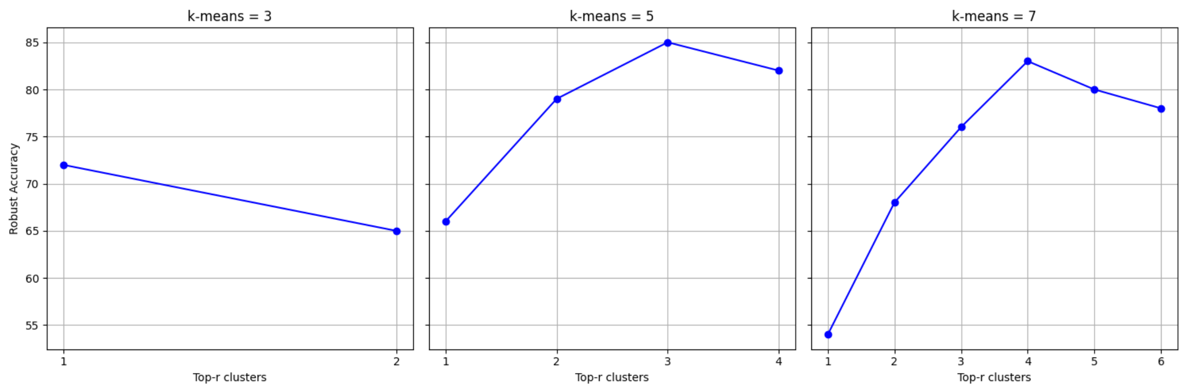


Figure 6.2: Robust accuracy vs. number of top-r clusters selected for inpainting using k-means clustering. Results are shown for $k = 3$ (left), $k = 5$ (middle), and $k = 7$ (right). Each plot demonstrates a peak in performance followed by a slight decrease, indicating the trade-off between removing adversarial perturbations and preserving benign features.

Thresholding: We first evaluated a thresholding approach by sorting pixel sensitivity values in decreasing order and selecting the top 5%, 10%, 20%, and 30% as potentially perturbed regions. The robust accuracy of our defense was then measured for each of these thresholds. Results showed a linear increase in robust accuracy up to the 20% threshold, followed by a sharp decrease at 30% (see Figure 6.3). This suggests that while masking a small percentage of the most sensitive pixels effectively removes adversarial perturbations, masking too large a region negatively impacts performance. This decrease likely results from the removal of benign image features important for correct classification. By masking beyond the 20% threshold, the inpainting process not only removes potential adversarial perturbations, but also eliminates essential contextual information, reducing the classifier's ability to make accurate predictions.

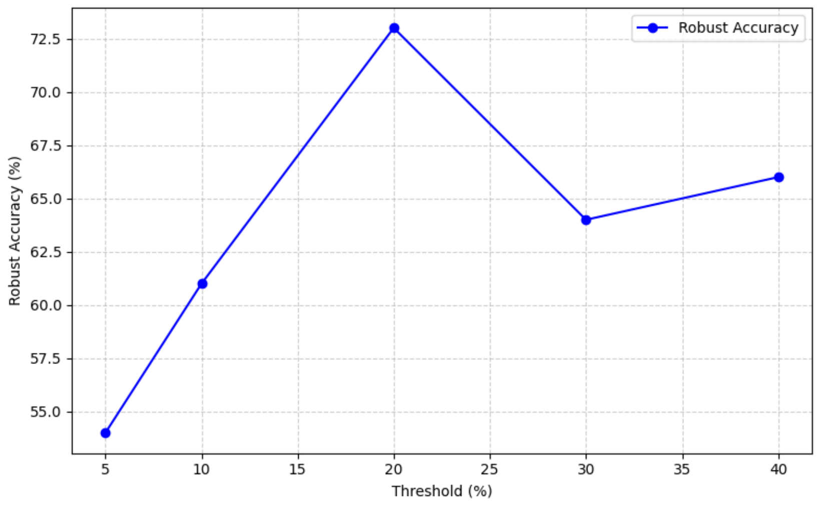


Figure 6.3: Robust accuracy vs. threshold percentage for mask generation using sensitivity-based thresholding. The plot shows that performance peaks at 20% and then decreases, suggesting that masking too many pixels removes important benign features crucial for accurate classification.

Clustering: Subsequently, we investigated a clustering-based approach using k-means with k values of 3, 5, and 7. Similar to the thresholding approach, we selected the top-r clusters based on their centroid sensitivity and calculated the robust accuracy for each configuration. Figure 6.2 presents the performance of these experiments. Each k value exhibits a peak in robust accuracy followed by a slight decrease as more clusters are selected for inpainting. This trend mirrors the observations from the thresholding approach, reinforcing the balance between removing adversarial perturbations and preserving benign features. Selecting too many clusters for inpainting results in the loss of crucial image information, ultimately diminishing the defense's effectiveness.

When comparing the two approaches, k-means clustering consistently yields better robust accuracy than thresholding. This superior performance can be attributed to the clustering algorithm's ability to capture the inherent structure of important image regions. By grouping pixels based on both sensitivity and spatial proximity, clustering potentially identifies perturbed regions that a rank-based thresholding approach might miss. Furthermore, clustering ensures that connected regions, rather than isolated superpixels, are targeted for inpainting, preserving contextual information and minimizing the disruption of benign features. This characteristic proves essential in maintaining the integrity of the image and facilitating accurate classification even after inpainting. Therefore, we conclude that clustering-based mask generation offers a more robust and effective approach for adversarial purification in MSID compared to simple thresholding techniques.

6.5. Evaluating imputation strategies for robust occlusion sensitivity maps

This research question investigated the impact of different imputation strategies on the robustness of MSID against adversarial attacks, specifically focusing on the generation of the occlusion sensitivity map. This map identifies important image regions for classification. We hypothesized that the method used to fill occluded regions during sensitivity analysis would significantly influence the robustness of the defense. Three imputation strategies were evaluated: (1) Zero-value imputation; (2) Histogram-based imputation and (3) Blurring-based imputation. These represent a set of approaches, from simple constant filling to more context-aware methods. Each imputation method was tested on images perturbed by a PGD attack with $\epsilon = 8/255$.

Our results demonstrate a clear difference in the robustness achieved with different imputation methods. Zero-value imputation resulted in a robust accuracy of 79%, while histogram-based imputation achieved 76%. Blurring-based imputation outperformed the others, achieving a robust accuracy of 86%. These findings are showed in the Figure 6.4.

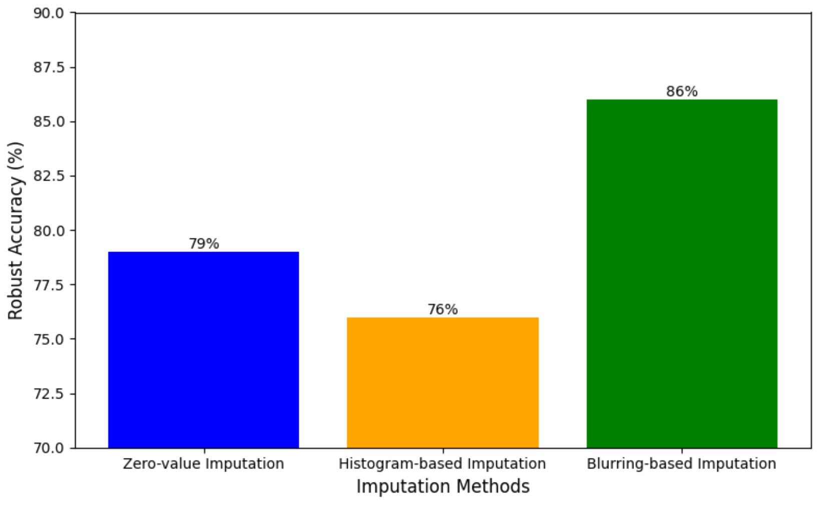


Figure 6.4: Robust accuracy of MSID under PGD attack (epsilon=8/255) using different imputation strategies during occlusion sensitivity map generation. Blurring-based imputation demonstrates superior robustness (86%) compared to zero-value (79%) and histogram-based (76%) imputation.

We think that blurring-based imputation's superior performance arises from its ability to preserve contextual information. The Gaussian blur, unlike the sharp changes introduced by zero-value or even the potentially harsh constant color fills of histogram-based imputation, hides features rather than erasing them. This minimizes the introduction of out-of-distribution (OOD) artifacts by maintaining contextual similarity in occluded areas. By smoothing out sharp transitions and preserving the overall image structure, blurring can bring adversarial examples closer to their ground-truth labels, potentially reversing adversarial properties.

In contrast, zero-value imputation, while simple, creates discontinuities in the image. These artificial boundaries introduce unnatural features that are not present in the original data distribution. Furthermore, the complete removal of information in the occluded region can lead to a loss of crucial contextual cues that might be necessary for accurate sensitivity maps generation.

Histogram-based imputation, while attempting to maintain some color consistency, suffers from a similar, although less extreme issue. While the filled regions are less harsh than zero-value imputation, the constant color blocks can still introduce unnatural patterns. Additionally, the random sampling from the histogram, while preserving some color characteristics, fails to capture the spatial structure and texture of the occluded region, further obstructing accurate sensitivity map generation.

Therefore, our findings suggest that preserving contextual information during occlusion is essential for generating a robust sensitivity map and consequently, for enhancing the robustness of MSID against adversarial attacks. Blurring-based imputation, by maintaining this contextual similarity and minimizing OOD artifacts, provides the most effective strategy for occlusion in this context.

7 Limitations and future work

7.1. Limitations

This work presents a novel approach to AP using multi-scale inpainting with DDPMs. Despite the promising results, several limitations should be acknowledged.

Computational cost: MSID relies on multi-scale superpixel segmentation, occlusion sensitivity map generation, and iterative DDPM inpainting. These processes, especially the DDPM component, are computationally expensive, potentially limiting the applicability of MSID in real-time or resource-constrained environments.

Dependence on pre-trained DDPMs: The effectiveness of MSID is tied to the quality of the pre-trained DDPM. Performance might vary depending on the dataset and architecture the DDPM was trained on, potentially requiring retraining or fine-tuning for optimal performance across different tasks. Furthermore, the availability of pre-trained DDPMs for specific datasets or domains can be a limitation.

Limited evaluation against adaptive attacks: While the current evaluation includes a gray-box scenario (as detailed in Chapter 5) and testing only against BPDA+EOT on CIFAR-10, a more comprehensive assessment against a wider set of adaptive attacks is important. The specific mechanisms of MSID, particularly the occlusion sensitivity maps, could potentially be exploited by adaptive adversaries to create stronger adversarial examples.

Hyperparameter Sensitivity: MSID introduces several hyperparameters related to superpixel segmentation, occlusion sensitivity analysis, DDPM inpainting, and artifact removal. The optimal values for these hyperparameters vary across datasets and attack types, requiring careful tuning. A more thorough investigation of hyperparameter sensitivity and potential automated tuning methods could further enhance the defense.

Limited comparison against diverse defense strategies: This work benchmarks MSID primarily against DM-based adversarial purification and AT. While demonstrating state-of-the-art performance within this subset of defenses, a broader evaluation covering other defense strategies, including certified defenses or non-diffusion based AP is necessary for a more comprehensive assessment of MSID's capabilities.

7.2. Future work

This work opens up some interesting research directions. One key area is boosting the speed of our method. Right now, the DDPM inpainting step is a computational bottleneck. We need to explore faster ways to sample from DDPMs or even entirely different inpainting methods that are less computationally intensive without compromising image quality. Improving how we approximate the occlusion sensitivity maps could also help speed things up. Furthermore, investigating alternative Explainable AI (XAI) techniques for guiding the inpainting process could lead to more efficient and targeted restorations.

We also need to make MSID more robust against adaptive attacks. Our current testing has not covered the full spectrum of adversarial attacks, particularly strong adaptive attacks

Setting the right hyperparameters for MSID can be tricky. Automating this process, would make the method more adaptable to different datasets and tasks.

Finally, we need a more comprehensive comparison of MSID against other defense strategies. This includes certified defenses and purification methods that do not rely on diffusion models. A more comprehensive comparison will help us understand MSID's strengths and weaknesses compared to other defenses.

8 Conclusion

This research paper introduced MSID, a novel adversarial purification defense leveraging the denoising capabilities of DDPMs and the explanatory power of occlusion sensitivity maps. Our approach, to the best of our knowledge, is the first to apply the use of DDPM inpainting for targeted perturbation removal, effectively neutralizing adversarial triggers while preserving benign image content. By integrating multi-scale occlusion analysis, MSID accurately identifies image regions that likely contain adversarial perturbation, guiding the DDPM inpainting process for precise attack mitigation.

Our extensive experiments across diverse datasets (CIFAR-10, Celeba-HQ, Imagenette and ImageNet) and classifier architectures (ResNet, WideResNet, MobileNet and ViT) demonstrate MSID's state-of-the-art performance. MSID surpasses existing AP and AT methods in robust accuracy, achieving gains against AutoAttack (up to 9.68% on CIFAR-10 and 33.28% on ImageNet) and PGD+EOT (up to 20.09% on CIFAR-10 and 27.7% on ImageNet). Importantly, MSID shows remarkable effectiveness against color-based attacks like cADV, achieving 64.84% robust accuracy on ImageNet and 75.19% on CIFAR-10, highlighting its ability to handle attacks beyond traditional L_p-norm perturbations. Furthermore, MSID demonstrates superior generalization to unseen attacks, outperforming state-of-the-art AT methods by up to 46.5%.

These findings underscore the potential of integrating XAI techniques with generative models for robust defense mechanisms. MSID's targeted inpainting approach offers a promising direction for developing defenses that are both effective and efficient.

In the following section, we provide answers to the research questions:

How can diffusion-based inpainting be leveraged to develop a more robust defense against adversarial attacks on image classifiers?

The thesis answers this by proposing MSID, which uses a pre-trained DDPM for inpainting. The method masks regions identified as likely to contain perturbations and then uses the DDPM to inpaint these regions, effectively restoring the image and removing the adversarial manipulations. The use of inpainting allows for targeted removal of perturbations while preserving benign image features.

Can a multi-scale approach to occlusion sensitivity mapping improve the identification and removal of adversarial perturbations?

The multi-scale approach, using superpixels at different scales, allows MSID to capture both fine-grained and coarse features, leading to more accurate identification of perturbed regions. The results demonstrate improved robustness compared to a single-scale approach. This targeted masking leads to more effective inpainting and better overall defense.

What methods can be used to generate a binary mask from the sensitivity map that accurately isolates regions that require inpainting for effective adversarial purification in MSID?

Evaluating thresholding and clustering methods for binary mask generation showed that k-means clustering consistently outperformed thresholding (see Figures 6.3 and 6.2). Clustering's ability to group pixels based on sensitivity and spatial proximity allows for more accurate identification of potentially perturbed regions while preserving benign features.

How does the choice of imputation strategy during occlusion sensitivity map generation influence the robustness of MSID against adversarial attacks?

The testing of different imputation strategies for the generation of occlusion sensitivity maps demonstrated that blurred-based imputation achieved the highest robust accuracy (86%) compared to zero-value (79%) and histogram-based (76%) imputation (see Figure 6.4). This superior performance is attributed to blurring's preservation of contextual information, minimizing out-of-distribution artifacts.

Does MSID provide better robustness against a wider range of adversarial attacks, including unseen attacks and color-based attacks, compared to existing state-of-the-art defenses?

The experimental results show that MSID achieves state-of-the-art robust accuracy against various attacks, including AutoAttack, PGD, strong adaptive attacks and unseen attacks, across different datasets and model architectures. Specifically, it demonstrates improved robustness against color-based attacks (cADV), a weakness of previous diffusion-based methods. The improved performance across a range of attacks suggests that MSID offers a robust defense mechanism.

References

[1] Radhakrishna Achanta et al. “SLIC superpixels compared to state-of-the-art superpixel methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2281. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.120.

[2] Maksym Andriushchenko and Nicolas Flammarion. “Understanding and Improving Fast Adversarial Training”. In: *Advances in Neural Information Processing Systems* 2020-December (July 2020). ISSN: 10495258. URL: https://arxiv.org/abs/2007.02617v2.

[3] Maksym Andriushchenko et al. “Square Attack: a query-efficient black-box adversarial attack via random search”. In: *Lecture Notes in Computer Science* 12368 LNCS (Nov. 2019), pp. 484–501. ISSN: 16113349. DOI: 10.1007/978-3-030-58592-1{_}29. URL: https://arxiv.org/abs/1912.00049v3.

[4] Daniel W. Apley and Jingyu Zhu. “Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models”. In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 82.4 (Dec. 2016), pp. 1059–1086. ISSN: 14679868. DOI: 10.1111/rssb.12377. URL: https://arxiv.org/abs/1612.08468v2.

[5] Anish Athalye, Nicholas Carlini, and David Wagner. “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples”. In: *International Conference on Machine Learning, 2018* 1 (Feb. 2018), pp. 436–448. URL: https://arxiv.org/abs/1802.00420v4.

[6] Anish Athalye et al. “Synthesizing Robust Adversarial Examples”. In: (July 2017). URL: http://arxiv.org/abs/1707.07397.

[7] Maximilian Augustin, Alexander Meinke, and Matthias Hein. “Adversarial Robustness on In- and Out-Distribution Improves Explainability”. In: *Lecture Notes in Computer Science* 12371 LNCS (Mar. 2020), pp. 228–245. ISSN: 16113349. DOI: 10.1007/978-3-030-58574-7{_}14. URL: https://arxiv.org/abs/2003.09461v2.

[8] Yutong Bai et al. “Are Transformers More Robust Than CNNs?” In: *Advances in Neural Information Processing Systems* 32 (Nov. 2021), pp. 26831–26843. ISSN: 10495258. URL: https://arxiv.org/abs/2111.05464v1.

[9] Anand Bhattad et al. “Unrestricted Adversarial Examples via Semantic Manipulation”. In: *International Conference on Learning Representations, 2020* (Apr. 2019). URL: https://arxiv.org/abs/1904.06347v2.

[10] Xiuli Bi et al. “ZeroPur: Succinct Training-Free Adversarial Purification”. In: (June 2024). URL: https://arxiv.org/abs/2406.03143v1.

[11] Stefan Blücher, Johanna Vielhaben, and Nils Strodthoff. “Decoupling Pixel Flipping and Occlusion Strategy for Consistent XAI Benchmarks”. In: *Transactions on Machine Learning Research* (Jan. 2024). URL: http://arxiv.org/abs/2401.06654.

[12] Qi Zhi Cai, Chang Liu, and Dawn Song. “Curriculum Adversarial Training”. In: *IJCAI International Joint Conference on Artificial Intelligence* 2018-July (May 2018), pp. 3740–3747. ISSN: 10450823. DOI: 10.24963/ijcai.2018/520. URL: https://arxiv.org/abs/1805.04807v1.

[13] Nicholas Carlini and David Wagner. “Towards Evaluating the Robustness of Neural Networks”. In: (Aug. 2016). URL: http://arxiv.org/abs/1608.04644.

[14] Nicholas Carlini et al. “(Certified!!) Adversarial Robustness for Free!” In: *11th International Conference on Learning Representations, ICLR 2023* (June 2022). URL: https://arxiv.org/abs/2206.10550v2.

[15] Huanran Chen et al. “Robust Classification via a Single Diffusion Model”. In: *Proceedings of Machine Learning Research* 235 (May 2023), pp. 6643–6665. ISSN: 26403498. URL: https://arxiv.org/abs/2305.15241v2.

[16] Sizhe Chen et al. *Adversarial Attack on Attackers: Post-Process to Mitigate Black-Box Score-Based Query Attacks*. Tech. rep.

[17] Hyungjin Chung et al. “Diffusion Posterior Sampling for General Noisy Inverse Problems”. In: *International Conference on Learning Representations* (Sept. 2022). URL: https://arxiv.org/abs/2209.14687v4.

[18] Joana C. Costa et al. “How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defenses”. In: *IEEE Access* 12 (May 2023), pp. 61113–61136. DOI: 10.1109/ACCESS.2024.3395118. URL: http://arxiv.org/abs/2305.10862%20http://dx.doi.org/10.1109/ACCESS.2024.3395118.

[19] Francesco Croce and Matthias Hein. *Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks*. Tech. rep. 2020.

[20] Francesco Croce et al. “Evaluating the Adversarial Robustness of Adaptive Test-time Defenses”. In: *Machine Learning Research* 162 (Feb. 2022), pp. 4421–4435. ISSN: 26403498. URL: https://arxiv.org/abs/2202.13711v2.

[21] Jiequan Cui et al. “Decoupled Kullback-Leibler Divergence Loss”. In: (May 2023). URL: https://arxiv.org/abs/2305.13948v3.

[22] Prafulla Dhariwal and Alex Nichol. “Diffusion Models Beat GANs on Image Synthesis”. In: *Advances in Neural Information Processing Systems* 11 (May 2021), pp. 8780–8794. ISSN: 10495258. URL: https://arxiv.org/abs/2105.05233v4.

[23] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. “advertorch v0.1: An Adversarial Robustness Toolbox based on PyTorch”. In: (Feb. 2019). URL: https://arxiv.org/abs/1902.07623v1.

[24] Hadi M Dolatabadi, Sarah Erfani, and Christopher Leckie. *ℓ ∞-Robustness and Beyond: Unleashing Efficient Adversarial Training*. Tech. rep.

[25] Hadi M. Dolatabadi, Sarah Erfani, and Christopher Leckie. “l∞-Robustness and Beyond: Unleashing Efficient Adversarial Training”. In: *Conference on Learning Representations, 2020* (Dec. 2021). URL: https://arxiv.org/abs/2112.00378v2.

[26] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations* (Oct. 2020). URL: https://arxiv.org/abs/2010.11929v2.

[27] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. “A study of the effect of JPG compression on adversarial images”. In: (Aug. 2016). URL: https://arxiv.org/abs/1608.00853v1.

[28] *fastai/imagenette: A smaller subset of 10 easily classified classes from Imagenet, and a little more French*. URL: https://github.com/fastai/imagenette.

[29] Reuben Feinman et al. “Detecting Adversarial Samples from Artifacts”. In: (Mar. 2017). URL: https://arxiv.org/abs/1703.00410v3.

[30] Thomas Fel et al. “Don’t Lie to Me! Robust and Efficient Explainability with Verified Perturbation Analysis”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2023-June (Feb. 2022), pp. 16153–16163. ISSN: 10636919. DOI: 10.1109/CVPR52729.2023.01550. URL: https://arxiv.org/abs/2202.07728v2.

[31] Ruth Fong and Andrea Vedaldi. “Interpretable Explanations of Black Boxes by Meaningful Perturbation”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-October (Apr. 2017), pp. 3449–3457. DOI: 10.1109/ICCV.2017.371. URL: http://arxiv.org/abs/1704.03296%20http://dx.doi.org/10.1109/ICCV.2017.371.

[32] Donald Geman and Chengda Yang. “Nonlinear Image Recovery with Half-Quadratic Regularization”. In: *IEEE Transactions on Image Processing* 4.7 (1995), pp. 932–946. ISSN: 19410042. DOI: 10.1109/83.392335.

[33] Amirata Ghorbani et al. “Towards Automatic Concept-based Explanations”. In: *Advances in Neural Information Processing Systems* 32 (Feb. 2019). ISSN: 10495258. URL: https://arxiv.org/abs/1902.03129v3.

[34] Partha Ghosh, Arpan Losalka, and Michael J Black. “Resisting Adversarial Attacks using Gaussian Mixture Variational Autoencoders”. In: (May 2018). URL: http://arxiv.org/abs/1806.00081.

[35] Alex Goldstein et al. “Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation”. In: *Journal of Computational and Graphical Statistics* 24.1 (Sept. 2013), pp. 44–65. ISSN: 15372715. DOI: 10.1080/10618600.2014.907095. URL: https://arxiv.org/abs/1309.6392v2.

[36] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: (Dec. 2014). URL: http://arxiv.org/abs/1412.6572.

[37] Sven Gowal et al. “Improving Robustness using Generated Data”. In: *Advances in Neural Information Processing Systems* 6 (Oct. 2021), pp. 4218–4233. ISSN: 10495258. URL: https://arxiv.org/abs/2110.09468v2.

[38] Sven Gowal et al. “Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples”. In: (Oct. 2020). URL: http://arxiv.org/abs/2010.03593.

[39] Chuan Guo et al. “Countering Adversarial Images using Input Transformations”. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (Oct. 2017). URL: https://arxiv.org/abs/1711.00117v3.

[40] J. A. Hartigan and M. A. Wong. “Algorithm AS 136: A K-Means Clustering Algorithm”. In: *Applied Statistics* 28.1 (1979), p. 100. ISSN: 00359254. DOI: 10.2307/2346830.

[41] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (Dec. 2015), pp. 770–778. ISSN: 10636919. DOI: 10.1109/CVPR.2016.90. URL: https://arxiv.org/abs/1512.03385v1.

[42] Mitch Hill, Jonathan Mitchell, and Song-Chun Zhu. “Stochastic Security: Adversarial Defense Using Long-Run Dynamics of Energy-Based Models”. In: (May 2020). URL: http://arxiv.org/abs/2005.13525.

[43] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising Diffusion Probabilistic Models”. In: (June 2020). URL: http://arxiv.org/abs/2006.11239.

[44] Uiwon Hwang et al. “PuVAE: A Variational Autoencoder to Purify Adversarial Examples”. In: (Mar. 2019). URL: http://arxiv.org/abs/1903.00585.

[45] Ioannis Kakogeorgiou and Konstantinos Karantzalos. “Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing”. In: *International Journal of Applied Earth Observation and Geoinformation* 103 (Dec. 2021), p. 102520. ISSN: 1569-8432. DOI: 10.1016/J.JAG.2021.102520.

[46] Tero Karras et al. “Progressive Growing of GANs for Improved Quality, Stability, and Variation”. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (Oct. 2017). URL: https://arxiv.org/abs/1710.10196v3.

[47] Bahjat Kawar et al. “Denoising Diffusion Restoration Models”. In: *Advances in Neural Information Processing Systems* 35 (Jan. 2022). ISSN: 10495258. URL: https://arxiv.org/abs/2201.11793v3.

[48] Been Kim et al. “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)”. In: *35th International Conference on Machine Learning, ICML 2018* 6 (Nov. 2017), pp. 4186–4195. URL: https://arxiv.org/abs/1711.11279v5.

[49] Hoki Kim. “Torchattacks: A PyTorch Repository for Adversarial Attacks”. In: (Sept. 2020). URL: https://arxiv.org/abs/2010.01950v3.

[50] Alex Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. In: (2009).

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems* 25 (2012). URL: `http://code.google.com/p/cuda-convnet/`.

[52] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. “Adversarial Machine Learning at Scale”. In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (Nov. 2016). URL: `https://arxiv.org/abs/1611.01236v2`.

[53] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. “Perceptual Adversarial Robustness: Defense Against Unseen Threat Models”. In: (June 2020). URL: `http://arxiv.org/abs/2006.12655`.

[54] Sebastian Lapuschkin et al. “Unmasking Clever Hans Predictors and Assessing What Machines Really Learn”. In: *Nature Communications* 10.1 (Feb. 2019). DOI: `10.1038/s41467-019-08987-4`. URL: `http://arxiv.org/abs/1902.10178%20http://dx.doi.org/10.1038/s41467-019-08987-4`.

[55] Minjong Lee and Dongwoo Kim. “Robust Evaluation of Diffusion-Based Adversarial Purification”. In: *Proceedings of the IEEE International Conference on Computer Vision* (Mar. 2023), pp. 134–144. ISSN: 15505499. DOI: `10.1109/ICCV51070.2023.00019`. URL: `https://arxiv.org/abs/2303.09051v3`.

[56] Simon Letzgus et al. “Toward Explainable AI for Regression Models”. In: *IEEE Signal Processing Magazine* 39.4 (Dec. 2021), pp. 40–58. DOI: `10.1109/MSP.2022.3153277`. URL: `http://arxiv.org/abs/2112.11407%20http://dx.doi.org/10.1109/MSP.2022.3153277`.

[57] Xiang Li and Shihao Ji. “Defense-VAE: A Fast and Accurate Defense against Adversarial Attacks”. In: *Communications in Computer and Information Science* 1168 CCIS (Dec. 2018), pp. 191–207. ISSN: 18650937. DOI: `10.1007/978-3-030-43887-6{\_}15`. URL: `https://arxiv.org/abs/1812.06570v3`.

[58] Yingwei Li et al. “Regional Homogeneity: Towards Learning Transferable Universal Adversarial Perturbations Against Defenses”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12356 LNCS (Apr. 2019), pp. 795–813. ISSN: 16113349. DOI: `10.1007/978-3-030-58621-8{\_}46`. URL: `https://arxiv.org/abs/1904.00979v2`.

[59] Fangzhou Liao et al. “Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Dec. 2017), pp. 1778–1787. ISSN: 10636919. DOI: `10.1109/CVPR.2018.00191`. URL: `https://arxiv.org/abs/1712.02976v2`.

[60] Guang Lin et al. *Adversarial Guided Diffusion Models for Adversarial Purification*.

[61] Guang Lin et al. “Adversarial Training on Purification (AToP): Advancing Both Robustness and Generalization”. In: (Jan. 2024). URL: `http://arxiv.org/abs/2401.16352`.

[62] Ziwei Liu et al. “Deep Learning Face Attributes in the Wild”. In: (Nov. 2014). URL: `http://arxiv.org/abs/1411.7766`.

[63] Jiajun Lu, Theerasit Issaranon, and David Forsyth. “SafetyNet: Detecting and Rejecting Adversarial Examples Robustly”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2017-October (Apr. 2017), pp. 446–454. ISSN: 15505499. DOI: `10.1109/ICCV.2017.56`. URL: `https://arxiv.org/abs/1704.00103v2`.

[64] Andreas Lugmayr et al. “RePaint: Inpainting using Denoising Diffusion Probabilistic Models”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2022-June (Jan. 2022), pp. 11451–11461. ISSN: 10636919. DOI: `10.1109/CVPR52688.2022.01117`. URL: `https://arxiv.org/abs/2201.09865v4`.

[65] Aleksander Madry et al. “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: (June 2017). URL: `http://arxiv.org/abs/1706.06083`.

[66] Dongyu Meng and Hao Chen. “MagNet: a Two-Pronged Defense against Adversarial Examples”. In: *Proceedings of the ACM Conference on Computer and Communications Security* (May 2017), pp. 135–147. ISSN: 15437221. DOI: `10.1145/3133956.3134057`. URL: `https://arxiv.org/abs/1705.09064v2`.

[67] Jan Hendrik Metzen et al. “On Detecting Adversarial Perturbations”. In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (Feb. 2017). URL: https://arxiv.org/abs/1702.04267v2.

[68] Seyed Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. “DeepFool: a simple and accurate method to fool deep neural networks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (Nov. 2015), pp. 2574–2582. ISSN: 10636919. DOI: 10.1109/CVPR.2016.282. URL: https://arxiv.org/abs/1511.04599v3.

[69] Konda Reddy Mopuri, Aditya Ganeshan, and R. Venkatesh Babu. “Generalizable Data-free Objective for Crafting Universal Adversarial Perturbations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.10 (Jan. 2018), pp. 2452–2465. ISSN: 19393539. DOI: 10.1109/TPAMI.2018.2861800. URL: https://arxiv.org/abs/1801.08092v3.

[70] Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. “Fast Feature Fool: A data independent approach to universal adversarial perturbations”. In: *British Machine Vision Conference 2017, BMVC 2017* (July 2017). DOI: 10.5244/c.31.30. URL: https://arxiv.org/abs/1707.05572v1.

[71] Dongbin Na, Sangwoo Ji, and Jong Kim. “Unrestricted Black-box Adversarial Attack Using GAN with Limited Queries”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13801 LNCS (Aug. 2022), pp. 467–482. ISSN: 16113349. DOI: 10.1007/978-3-031-25056-9{_}30. URL: https://arxiv.org/abs/2208.11613v1.

[72] Muzammal Naseer et al. “A Self-supervised Approach for Adversarial Robustness”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (June 2020), pp. 259–268. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00034. URL: https://arxiv.org/abs/2006.04924v1.

[73] Alex Nichol and Prafulla Dhariwal. “Improved Denoising Diffusion Probabilistic Models”. In: *Proceedings of Machine Learning Research* 139 (Feb. 2021), pp. 8162–8171. ISSN: 26403498. URL: https://arxiv.org/abs/2102.09672v1.

[74] Maria-Irina Nicolae et al. “Adversarial Robustness Toolbox v1.0.0”. In: (July 2018). URL: https://arxiv.org/abs/1807.01069v4.

[75] Weili Nie et al. “Diffusion Models for Adversarial Purification”. In: (May 2022). URL: http://arxiv.org/abs/2205.07460.

[76] Tianyu Pang et al. “Robustness and Accuracy Could Be Reconcilable by (Proper) Definition”. In: *Proceedings of Machine Learning Research* 162 (Feb. 2022), pp. 17258–17277. ISSN: 26403498. URL: https://arxiv.org/abs/2202.10103v2.

[77] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems* 32 (Dec. 2019). ISSN: 10495258. URL: https://arxiv.org/abs/1912.01703v1.

[78] Omid Poursaeed et al. *Robustness and Generalization via Generative Adversarial Training*. Tech. rep.

[79] Han Qiu et al. “An Efficient Preprocessing-Based Approach to Mitigate Advanced Adversarial Attacks”. In: *IEEE Transactions on Computers* 73.3 (Mar. 2024), pp. 645–655. ISSN: 15579956. DOI: 10.1109/TC.2021.3076826.

[80] Edward Raff et al. “Barrage of random transforms for adversarially robust defense”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (June 2019), pp. 6521–6530. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00669.

[81] Jonas Rauber, Wieland Brendel, and Matthias Bethge. “Foolbox: A Python toolbox to benchmark the robustness of machine learning models”. In: (July 2017). URL: https://arxiv.org/abs/1707.04131v3.

[82] Sylvestre-Alvise Rebuffi et al. “Fixing Data Augmentation to Improve Adversarial Robustness”. In: (Mar. 2021). URL: https://arxiv.org/abs/2103.01946v2.

[83] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9351 (2015), pp. 234–241. ISSN: 1611-3349. DOI: `10.1007/978-3-319-24574-4{\_}28`. URL: `https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28`.

[84] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: (Sept. 2014). URL: `https://arxiv.org/abs/1409.0575v3`.

[85] Hadi Salman et al. “Do Adversarially Robust ImageNet Models Transfer Better?” In: *Advances in Neural Information Processing Systems* 2020-December (July 2020). ISSN: 10495258. URL: `https://arxiv.org/abs/2007.08489v2`.

[86] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. “Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models”. In: (May 2018). URL: `http://arxiv.org/abs/1805.06605`.

[87] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Jan. 2018), pp. 4510–4520. ISSN: 10636919. DOI: `10.1109/CVPR.2018.00474`. URL: `https://arxiv.org/abs/1801.04381v4`.

[88] Lukas Schott et al. “Towards the first adversarially robust neural network model on MNIST”. In: (May 2018). URL: `http://arxiv.org/abs/1805.09190`.

[89] Vikash Sehwag et al. “Robust Learning Meets Generative Models: Can Proxy Distributions Improve Adversarial Robustness?” In: *ICLR 2022 - 10th International Conference on Learning Representations* (Apr. 2021). URL: `https://arxiv.org/abs/2104.09425v3`.

[90] Changhao Shi, Chester Holtz, and Gal Mishne. “Online Adversarial Purification based on Self-Supervision”. In: (Jan. 2021). URL: `http://arxiv.org/abs/2101.09387`.

[91] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models”. In: *ICLR 2021 - 9th International Conference on Learning Representations* (Oct. 2020). URL: `https://arxiv.org/abs/2010.02502v4`.

[92] Yang Song and Stefano Ermon. “Generative Modeling by Estimating Gradients of the Data Distribution”. In: (July 2019). URL: `http://arxiv.org/abs/1907.05600`.

[93] Yang Song et al. “PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples”. In: (Oct. 2017). URL: `http://arxiv.org/abs/1710.10766`.

[94] Yang Song et al. “Score-Based Generative Modeling through Stochastic Differential Equations”. In: (Nov. 2020). URL: `http://arxiv.org/abs/2011.13456`.

[95] Vignesh Srinivasan et al. “Robustifying Models Against Adversarial Attacks by Langevin Dynamics”. In: (May 2018). URL: `http://arxiv.org/abs/1805.12017`.

[96] David Stutz, Matthias Hein, and Bernt Schiele. “Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks”. In: *37th International Conference on Machine Learning, ICML 2020* PartF168147-12 (Oct. 2019), pp. 9092–9103. URL: `https://arxiv.org/abs/1910.06259v4`.

[97] David Stutz, Alexander Hermans, and Bastian Leibe. “Superpixels: An evaluation of the state-of-the-art”. In: *Computer Vision and Image Understanding* 166 (Jan. 2018), pp. 1–27. ISSN: 1077-3142. DOI: `10.1016/J.CVIU.2017.03.007`.

[98] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. “One pixel attack for fooling deep neural networks”. In: *IEEE Transactions on Evolutionary Computation* 23.5 (Oct. 2017), pp. 828–841. DOI: `10.1109/TEVC.2019.2890858`. URL: `http://arxiv.org/abs/1710.08864%20http://dx.doi.org/10.1109/TEVC.2019.2890858`.

[99] Christian Szegedy et al. “Intriguing properties of neural networks”. In: (Dec. 2013). URL: `http://arxiv.org/abs/1312.6199`.

[100] Iuliana Tabian, Hailing Fu, and Zahra Sharif Khodaei. “A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures”. In: *Sensors 2019, Vol. 19, Page 4933* 19.22 (Nov. 2019), p. 4933. ISSN: 1424-8220. DOI: 10.3390/S19224933. URL: https://www.mdpi.com/1424-8220/19/22/4933/htm%20https://www.mdpi.com/1424-8220/19/22/4933.

[101] Jihoon Tack et al. “Consistency Regularization for Adversarial Robustness”. In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022* 36 (Mar. 2021), pp. 8414–8422. ISSN: 2159-5399. DOI: 10.1609/aaai.v36i8.20817. URL: https://arxiv.org/abs/2103.04623v3.

[102] Florian Tramèr, Nicholas Carlini, and Wieland Brendel. *On Adaptive Attacks to Adversarial Example Defenses*. Tech. rep. URL: https://github.com/wielandbrendel/adaptive_attacks_paper..

[103] Florian Tramèr et al. “On Adaptive Attacks to Adversarial Example Defenses”. In: *Advances in Neural Information Processing Systems* 2020-December (Feb. 2020). ISSN: 10495258. URL: https://arxiv.org/abs/2002.08347v2.

[104] Jonathan Uesato et al. “Adversarial Risk and the Dangers of Evaluating Against Weak Attacks”. In: (Feb. 2018). URL: http://arxiv.org/abs/1802.05666.

[105] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems* 2017-December (June 2017), pp. 5999–6009. ISSN: 10495258. URL: https://arxiv.org/abs/1706.03762v7.

[106] Bas H. M. van der Velden et al. “Explainable artificial intelligence (XAI) in deep learning-based medical image analysis”. In: *Medical Image Analysis* 79 (July 2021). DOI: 10.1016/j.media.2022.102470. URL: http://arxiv.org/abs/2107.10912%20http://dx.doi.org/10.1016/j.media.2022.102470.

[107] *Vision Transformers - by Cameron R. Wolfe, Ph.D.* URL: https://cameronrwolfe.substack.com/p/vision-transformers.

[108] Stefan van der Walt et al. “scikit-image: Image processing in Python”. In: *PeerJ* 2014.1 (July 2014). DOI: 10.7717/peerj.453. URL: http://arxiv.org/abs/1407.6245%20http://dx.doi.org/10.7717/peerj.453.

[109] Haotao Wang et al. “Once-for-All Adversarial Training: In-Situ Tradeoff between Robustness and Accuracy for Free”. In: *Advances in Neural Information Processing Systems* 2020-December (Oct. 2020). ISSN: 10495258. URL: https://arxiv.org/abs/2010.11828v2.

[110] Jinyi Wang et al. “Guided Diffusion Model for Adversarial Purification”. In: (May 2022). URL: http://arxiv.org/abs/2205.14969.

[111] Yisen Wang et al. “On the Convergence and Robustness of Adversarial Training”. In: *36th International Conference on Machine Learning, ICML 2019* 2019-June (Dec. 2021), pp. 11426–11438. URL: https://arxiv.org/abs/2112.08304v2.

[112] Zekai Wang et al. “Better Diffusion Models Further Improve Adversarial Training”. In: *Proceedings of Machine Learning Research* 202 (Feb. 2023), pp. 36246–36263. ISSN: 26403498. URL: https://arxiv.org/abs/2302.04638v2.

[113] Eric Wong, Leslie Rice, and J. Zico Kolter. “Fast is better than free: Revisiting adversarial training”. In: (Jan. 2020). URL: http://arxiv.org/abs/2001.03994.

[114] Baoyuan Wu et al. “Defenses in Adversarial Machine Learning: A Survey”. In: (Dec. 2023). URL: https://arxiv.org/abs/2312.08890v1.

[115] Dongxian Wu, Shu-tao Xia, and Yisen Wang. “Adversarial Weight Perturbation Helps Robust Generalization”. In: (Apr. 2020). URL: http://arxiv.org/abs/2004.05884.

[116] Chaowei Xiao et al. “DensePure: Understanding Diffusion Models towards Adversarial Robustness”. In: *11th International Conference on Learning Representations, ICLR 2023* (Nov. 2022). URL: https://arxiv.org/abs/2211.00322v1.

[117] Chaowei Xiao et al. “Spatially Transformed Adversarial Examples”. In: (Jan. 2018). URL: http://arxiv.org/abs/1801.02612.

[118] Jie Xiao et al. *DREAMCLEAN: RESTORING CLEAN IMAGE USING DEEP DIFFUSION PRIOR*. Tech. rep.

[119] Cihang Xie et al. “Mitigating Adversarial Effects Through Randomization”. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (Nov. 2017). URL: `https://arxiv.org/abs/1711.01991v3`.

[120] Han Xu et al. “To be Robust or to be Fair: Towards Fairness in Adversarial Training”. In: *Proceedings of Machine Learning Research* 139 (Oct. 2020), pp. 11492–11501. ISSN: 26403498. URL: `https://arxiv.org/abs/2010.06121v2`.

[121] Weilin Xu, David Evans, and Yanjun Qi. “Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks”. In: *25th Annual Network and Distributed System Security Symposium, NDSS 2018* (Apr. 2017). DOI: `10.14722/ndss.2018.23198`. URL: `http://arxiv.org/abs/1704.01155%20http://dx.doi.org/10.14722/ndss.2018.23198`.

[122] Yuzhe Yang et al. “ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation”. In: (May 2019). URL: `http://arxiv.org/abs/1905.11971`.

[123] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. “Adversarial purification with Score-based generative models”. In: *Proceedings of Machine Learning Research* 139 (June 2021), pp. 12062–12072. ISSN: 26403498. URL: `https://arxiv.org/abs/2106.06041v1`.

[124] Sergey Zagoruyko and Nikos Komodakis. “Wide Residual Networks”. In: *British Machine Vision Conference 2016, BMVC 2016* 2016-September (May 2016), pp. 1–87. DOI: `10.5244/C.30.87`. URL: `https://arxiv.org/abs/1605.07146v4`.

[125] Matthew D Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks”. In: (Nov. 2013). URL: `http://arxiv.org/abs/1311.2901`.

[126] Hongyang Zhang et al. “Theoretically Principled Trade-off between Robustness and Accuracy”. In: (Jan. 2019). URL: `http://arxiv.org/abs/1901.08573`.

[127] Jiawei Zhang et al. “DiffSmooth: Certifiably Robust Learning via Diffusion Models and Local Smoothing”. In: *32nd USENIX Security Symposium, USENIX Security 2023* 7 (Aug. 2023), pp. 4787–4804. URL: `https://arxiv.org/abs/2308.14333v1`.

[128] Jingfeng Zhang et al. “Geometry-aware Instance-reweighted Adversarial Training”. In: (Oct. 2020). URL: `http://arxiv.org/abs/2010.01736`.

[129] Kai Zhang et al. “Plug-and-Play Image Restoration with Deep Denoiser Prior”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (Aug. 2020), pp. 6360–6376. ISSN: 19393539. DOI: `10.1109/TPAMI.2021.3088914`. URL: `https://arxiv.org/abs/2008.13751v2`.

[130] Yutong Zhang et al. “A Review of Adversarial Attacks in Computer Vision”. In: (Aug. 2023). URL: `https://arxiv.org/abs/2308.07673v1`.

[131] Zhihao Zheng and Pengyu Hong. “Robust Detection of Adversarial Attacks by Modeling the Intrinsic Properties of Deep Neural Networks”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[132] Yuanzhi Zhu et al. “Denoising Diffusion Models for Plug-and-Play Image Restoration”. In: (May 2023). URL: `http://arxiv.org/abs/2305.08995`.

A
Pre-trained Model Details

This appendix details the architectures and training parameters of the pre-trained DDPM models used in our experiments. We provide configurations for CIFAR-10, Celeba-HQ, ImageNet and ImageNette. For each dataset, we specify the diffusion parameters, training hyperparameters, and model-specific settings.

A.1. DDPM Implementations

A.1.1. CIFAR-10

The pre-trained model is available for download at this link.

Table A.1: CIFAR-10 DDPM Diffusion Parameters

Parameter	**Value**	**Description**
Timesteps	1000	Number of diffusion timesteps.
Beta Start	0.0001	Starting value for the noise schedule.
Beta End	0.02	Ending value for the noise schedule.
Beta Schedule	linear	Type of noise schedule used (linear in this case).
Model Mean Type	eps	How the model predicts the mean (epsilon prediction).
Model Variance Type	fixed-large	Method for handling the variance during training.
Loss Type	mse	Loss function used (mean squared error).

A.1.2. Celeba-HQ

The pre-trained model is available for download at this link.

Table A.2: CIFAR-10 DDPM Train Parameters

Parameter	Value	Description
Learning Rate	2e-4	Initial learning rate for the optimizer.
Batch Size	128	Batch size used during training.
Gradient Clipping Norm	1.0	Maximum norm for gradient clipping.
Epochs	2040	Number of training epochs.
Warmup Steps	5000	Number of warmup steps for the learning rate scheduler.
Use EMA	true	Whether exponential moving average of weights is used.
EMA Decay	0.9999	Decay rate for the exponential moving average.

Table A.3: CIFAR-10 DDPM Model Parameters

Parameter	Value	Description
Image Size	32	Size of the input image.
Number of Heads	4	Number of attention heads used in the attention blocks.
Attention Resolutions	”2”	Indicates at which image resolutions attention is applied.
Number of Upsample Heads	-1	Use the same number of heads for upsampling as for downsampling.
Use Scale-Shift Norm	True	Whether to use scale-shift normalization in attention layers.
ResBlock UpDown	True	Whether to use up/down sampling in residual blocks.
Use FP16	False	Indicates whether to use FP16 precision
Input Channels	3	Number of input channels (for RGB images).
Hidden Channels	128	Number of channels in the initial hidden layer of the U-Net.
Channel Multipliers	[1, 2, 2, 2]	Multipliers for the number of channels at each level of the U-Net.
Number of Residual Blocks	2	Number of residual blocks per U-Net level.
Attention Blocks	[false, true, false, false]	Indicates whether attention is applied at each level of the U-Net (true for the second level in this case).
Dropout Rate	0.1	Dropout rate used during training.

Table A.4: Celeba-HQ DDPM Diffusion Parameters

Parameter	**Value**	**Description**
Timesteps	1000	Number of diffusion timesteps.
Beta Start	0.0001	Starting value for the noise schedule.
Beta End	0.02	Ending value for the noise schedule.
Beta Schedule	linear	Type of noise schedule used (linear in this case).
Model Mean Type	eps	How the model predicts the mean (epsilon prediction).
Model Variance Type	fixed-small	Method for handling the variance during training.
Loss Type	mse	Loss function used (mean squared error).

Table A.5: Celeba-HQ DDPM Train Parameters

Parameter	**Value**	**Description**
Learning Rate	2e-5	Initial learning rate for the optimizer.
Batch Size	64	Batch size used during training.
Gradient Clipping Norm	1.0	Maximum norm for gradient clipping.
Epochs	1200	Number of training epochs.
Warmup Steps	5000	Number of warmup steps for the learning rate scheduler.
Use EMA	true	Whether exponential moving average of weights is used.
EMA Decay	0.9999	Decay rate for the exponential moving average.

Table A.6: Celeba-HQ DDPM Model Parameters

Parameter	**Value**	**Description**
Image Size	256	Size of the input image.
Number of Heads	4	Number of attention heads used in the attention blocks.
Attention Resolutions	”16”	Indicates at which image resolutions attention is applied.
Number of Upsample Heads	-1	Use the same number of heads for upsampling as for downsampling.
Use Scale-Shift Norm	True	Whether to use scale-shift normalization in attention layers.
ResBlock UpDown	True	Whether to use up/down sampling in residual blocks.
Use FP16	False	Indicates whether to use FP16 precision
Input Channels	3	Number of input channels (for RGB images).
Hidden Channels	128	Number of channels in the initial hidden layer of the U-Net.
Channel Multipliers	[1, 1, 2, 2, 4, 4]	Multipliers for the number of channels at each level of the U-Net.
Number of Residual Blocks	2	Number of residual blocks per U-Net level.
Attention Blocks	[false, false, false, false, true, false]	Indicates whether attention is applied at each level of the U-Net (true for the fifth level in this case).
Dropout Rate	0.0	Dropout rate used during training.

A.1.3. Imagenet

Given that ImageNette is a subset of ImageNet, these parameters remain valid and are utilized consistently across both datasets. The pre-trained model is available for download at this link.

Table A.7: Imagenet DDPM Model Parameters

Parameter	Value	Description
Image Size	256	Size of the input image.
Number of Heads	4	Number of attention heads used in the attention blocks.
Attention Resolutions	”32,16,8”	Indicates at which image resolutions attention is applied.
Number of Upsample Heads	-1	Use the same number of heads for upsampling as for downsampling.
Use Scale-Shift Norm	True	Whether to use scale-shift normalization in attention layers.
ResBlock UpDown	True	Whether to use up/down sampling in residual blocks.
Use FP16	False	Indicates whether to use FP16 precision
Input Channels	3	Number of input channels (for RGB images).
Hidden Channels	256	Number of channels in the initial hidden layer of the U-Net.
Channel Multipliers	[1, 1, 2, 2, 4, 4]	Multipliers for the number of channels at each level of the U-Net.
Number of Residual Blocks	2	Number of residual blocks per U-Net level.
Dropout Rate	0.0	Dropout rate used during training.

Table A.8: Imagenet DDPM Train Parameters

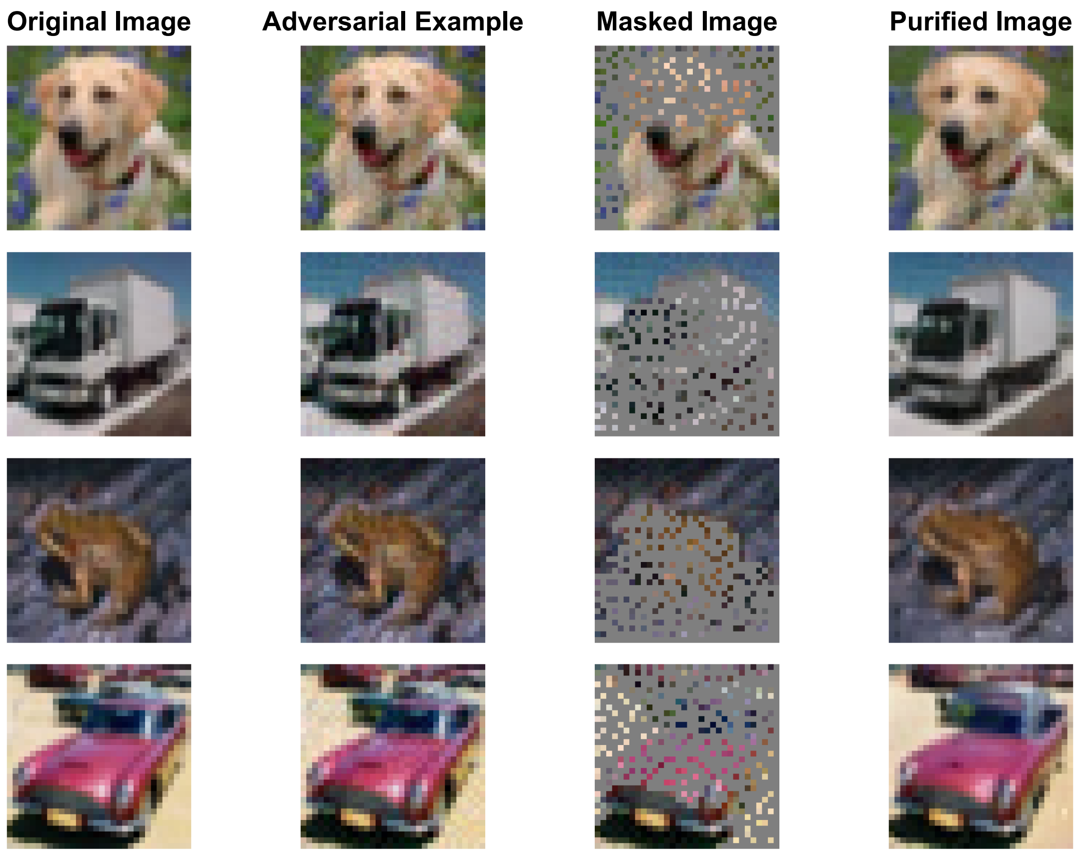
Parameter	Value	Description
Learning Rate	1e-4	Initial learning rate for the optimizer.
Batch Size	256	Batch size used during training.
Gradient Clipping Norm	1.0	Maximum norm for gradient clipping.
Epochs	1980	Number of training epochs.

Table A.9: Imagenet Diffusion Parameters

Parameter	Value	Description
Timesteps	1000	Number of diffusion timesteps.
Beta Start	0.0001	Starting value for the noise schedule.
Beta End	0.02	Ending value for the noise schedule.
Beta Schedule	linear	Type of noise schedule used (linear in this case).
Model Mean Type	eps	How the model predicts the mean (epsilon prediction).
Model Variance Type	fixed-small	Method for handling the variance during training.
Loss Type	mse	Loss function used (mean squared error).

B
Image samples

This appendix provides a visual showcase of the image purification process described in Chapter 4. Here, we present a collection of examples showing the effects of our method on adversarially perturbed images from the CIFAR-10, Imagnette and Celeba-HQ datasets. Each row displays the original image, its corresponding adversarial counterpart (generated using the Autoattack attack), the masked image highlighting the regions identified for purification, and finally, the resulting purified image after applying our MSID technique. These images offer insights into the effectiveness of our approach in recovering the original image content while purifying the adversarial perturbations. This visual inspection allows for a qualitative assessment of the performance of our method and complements the quantitative results presented in the main part of the thesis.



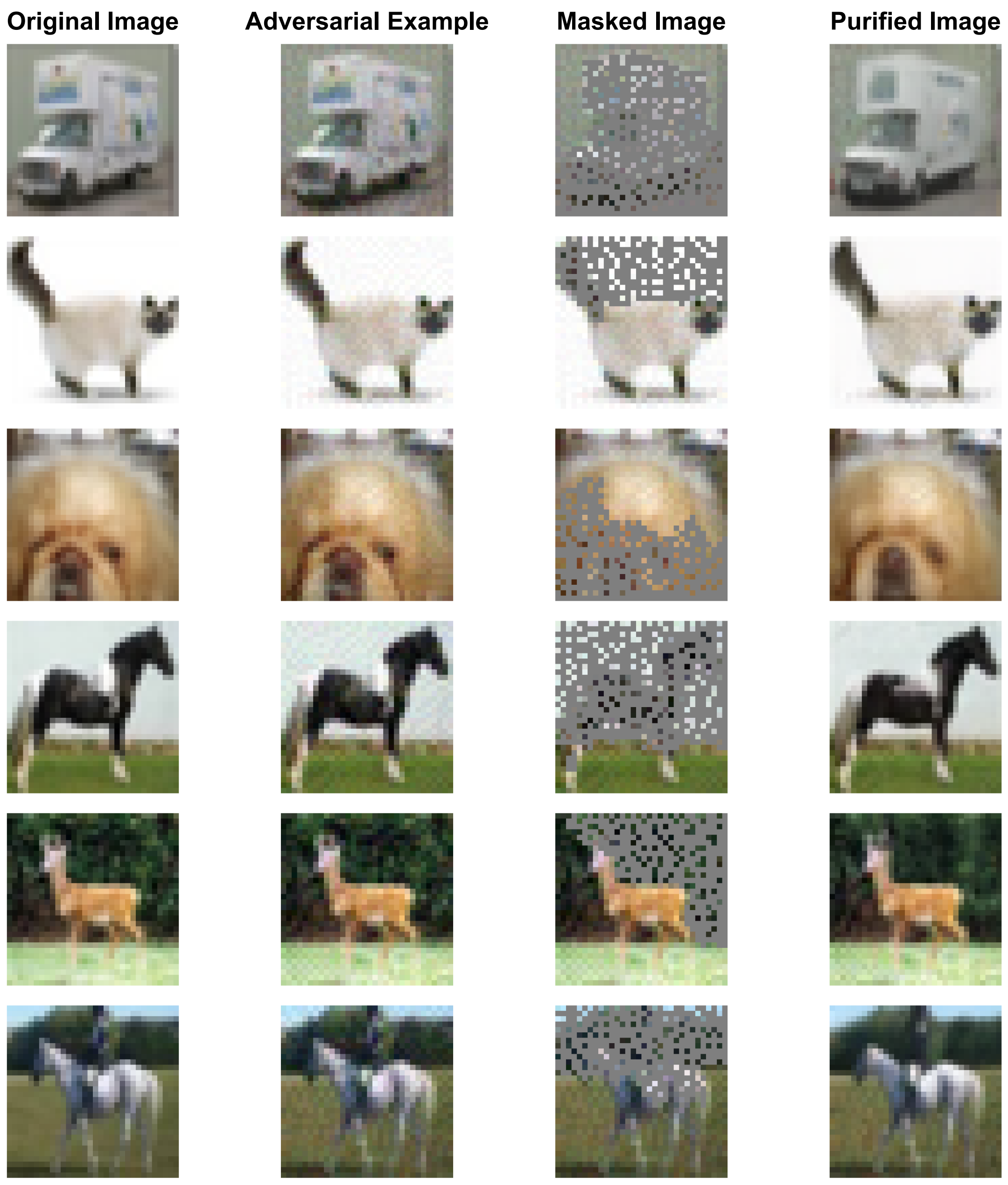
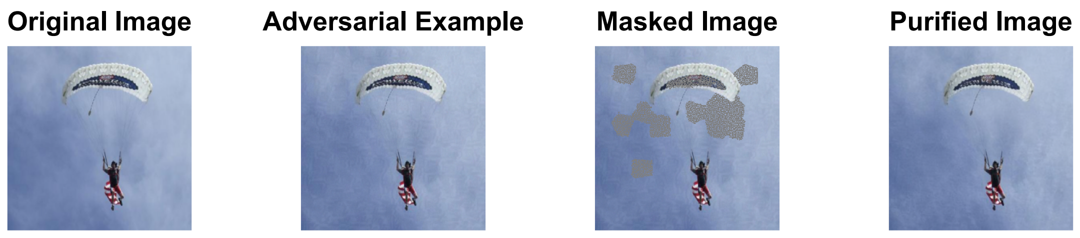
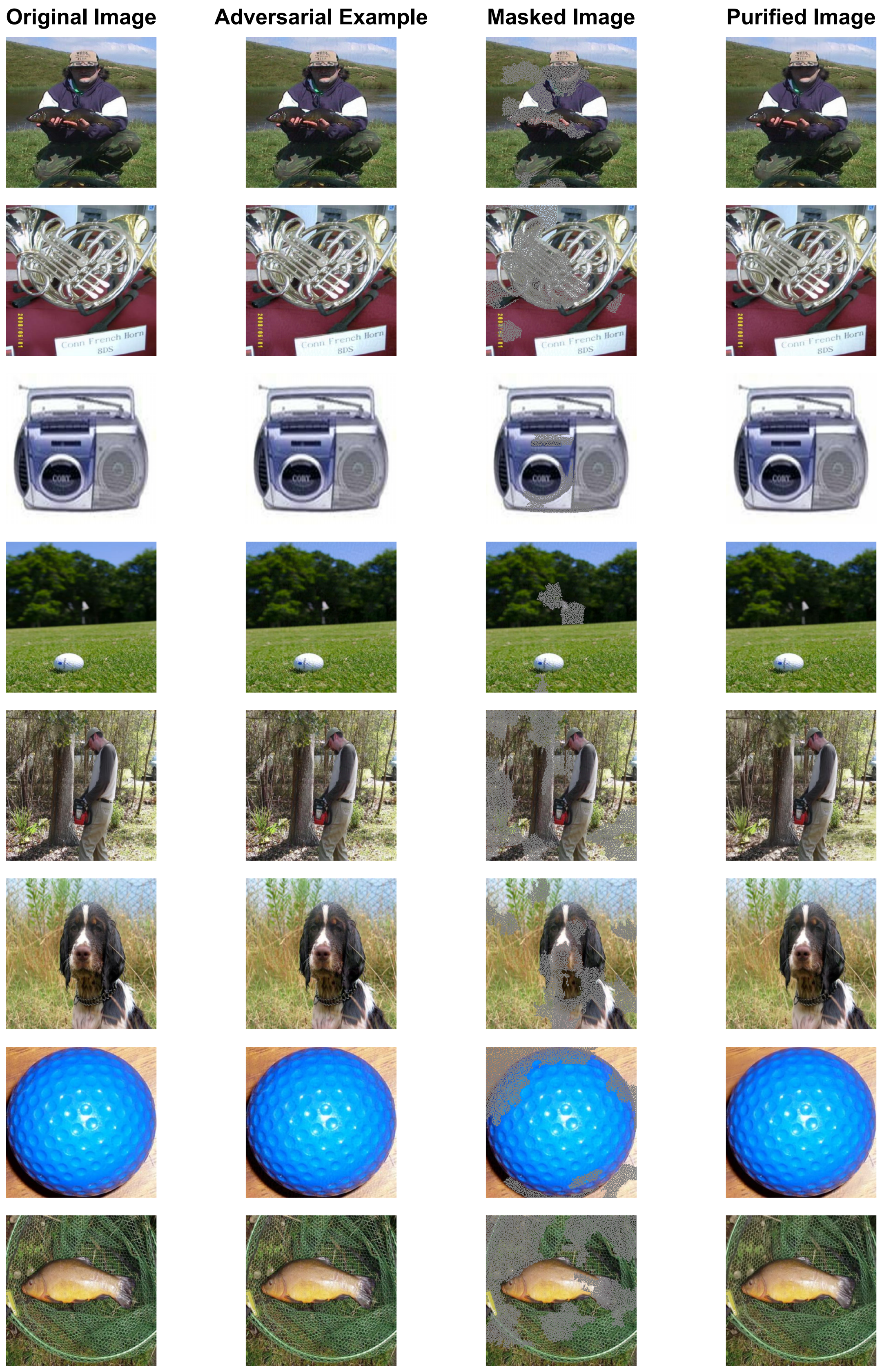


Table B.1: Purification of adversarial examples from the CIFAR-10 dataset. The original image is shown alongside its adversarially perturbed version (generated using the Autoattack attack), the masked image and the final purified image after applying MSID.




Original Image
Adversarial Example
Masked Image
Purified Image

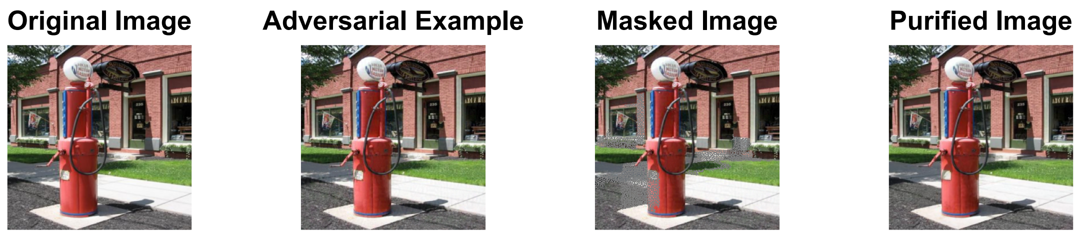
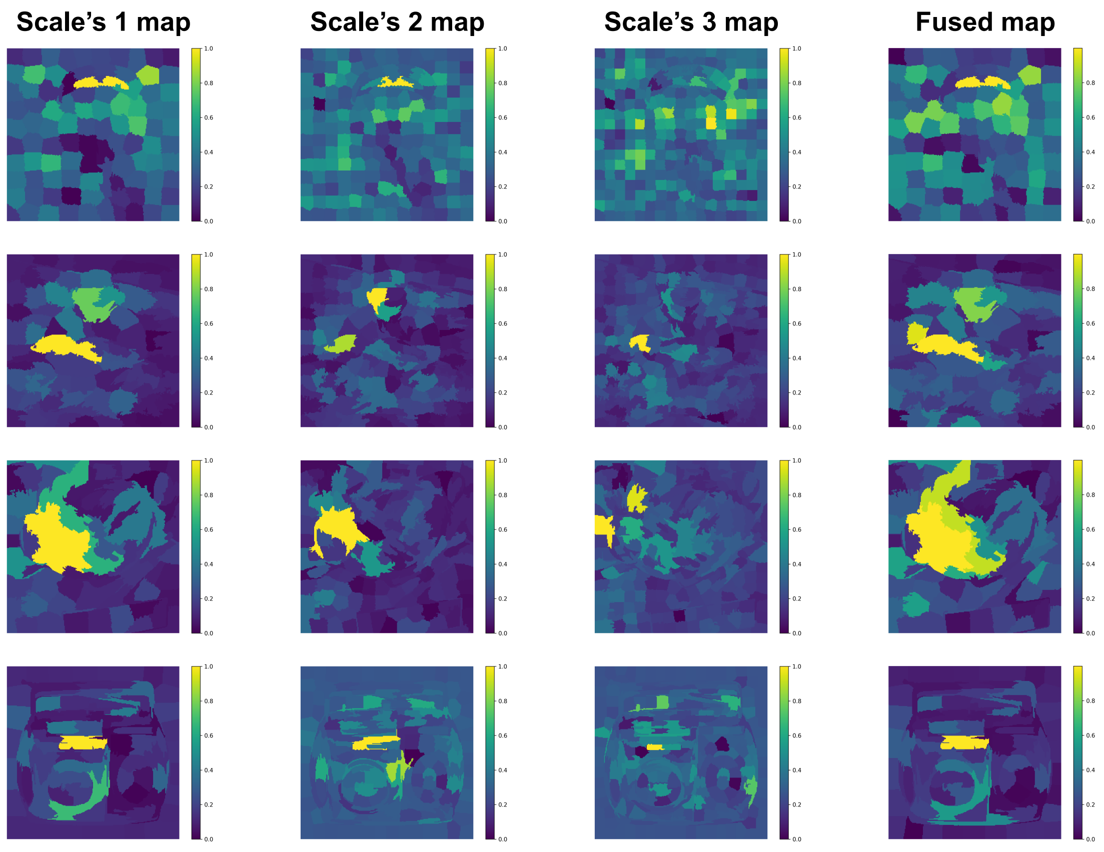


Table B.2: Purification of adversarial examples from the Imagenette dataset. The original image is shown alongside its adversarially perturbed version (generated using the Autoattack attack), the masked image and the final purified image after applying MSID.

C
Visualisation of occlusion sensitivity across scales

This appendix provides a visual exploration of the multi-scale sensitivity maps used in the MSID. As described in Chapter 4, MSID leverages occlusion sensitivity maps at different scales to identify sensitive regions for image classification. Here, we present visualizations of these maps at various levels of granularity, demonstrating how they capture different aspects of image. Furthermore, we visualize the fused sensitivity map, which combines these multi-scale maps into a single map, as detailed in Section 4.3.4. This visualization provides a deeper understanding of the hierarchical analysis performed by MSID. The visualizations illustrate how the fused map integrates information across scales, leading to a more precise identification of adversarial perturbations while preserving benign image features.



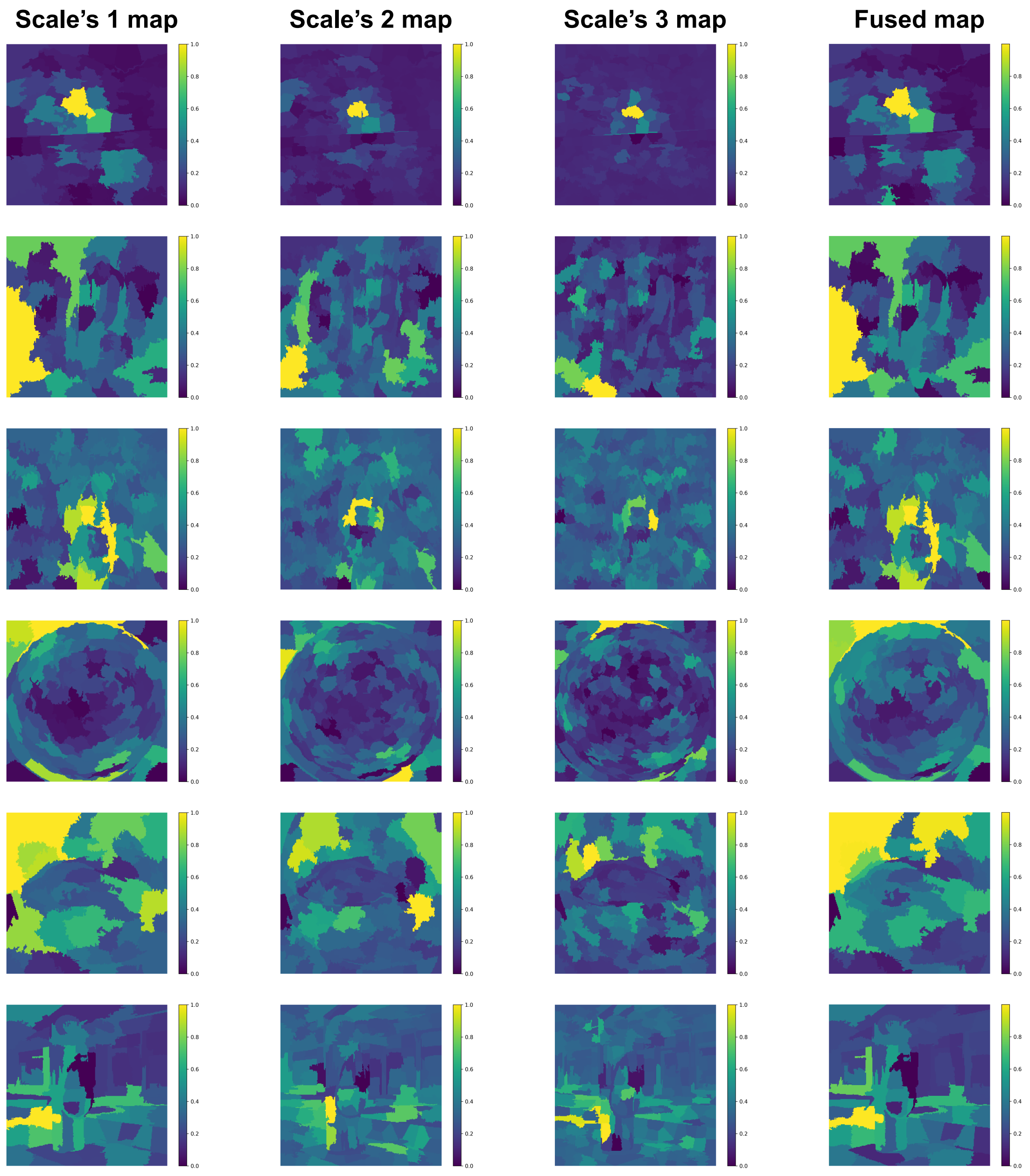


Table C.1: Occlusion sensitivity maps generated at different superpixel scales, along with the final fused map. These maps correspond to the image samples presented in Appendix B.

D
Source Code

This appendix provides the source code for the key components of the MSID presented in Chapter 4.

```python
def get_occlusion_map(img, model, n_pixels):
    """Generates an occlusion sensitivity map.

    Calculates the sensitivity of the model's prediction to the occlusion of
    individual superpixels in the image.

    Args:
        img: The input image as a tensor.
        model: The classifier model.
        n_pixels: Number of superpixels.

    Returns:
        A tuple containing the normalized occlusion heatmap and the superpixel segmentation.
    """
    # Obtain the baseline prediction and the predicted class label.
    baseline_prediction, label = get_occlusion_prediction(img, model)
    baseline_prediction = baseline_prediction[0, label]

    # Convert the input tensor image to a NumPy array.
    np_img_original = to_np_img(img)

    # Generate superpixel segmentation.
    superpixels = get_superpixels(np_img_original, n_pixels=n_pixels)
    unique_superpixels = np.unique(superpixels)

    # Initialize the occlusion sensitivity map.
    occlusion_heatmap = np.zeros_like(superpixels, dtype=np.float32)

    # Iterate over each superpixel.
    for superpixel_label in unique_superpixels:
        np_img = np_img_original.copy()

        # Create a mask for the current superpixel.
        superpixel_mask = (superpixels == superpixel_label)

        # Extract the pixel values of the current superpixel.
        superpixel_region = np_img[superpixel_mask]

        # Calculate the imputation value.
        impute_value = get_impute_value(superpixel_region, type="blur")

        # Occlude the superpixel with the imputed value.
        np_img[superpixel_mask] = impute_value

        # Convert the NumPy array back to a tensor.
        tensor_img = to_tensor_img(np_img)

        # Obtain the prediction for the occluded image.
        occluded_prediction, _ = get_occlusion_prediction(tensor_img, model)
        occluded_prediction = occluded_prediction[0, label]

        # Calculate the sensitivity as the difference between baseline and occluded predictions.
        sensitivity = baseline_prediction - occluded_prediction
        sensitivity = sensitivity.cpu().detach().numpy()

        # Update the sensitivity map with the calculated sensitivity.
        occlusion_heatmap[superpixel_mask] = sensitivity

    # Normalize the heatmap and return it along with the superpixel segmentation.
    return normalize_heatmap(occlusion_heatmap), superpixels
```

Listing D.1: Generating Occlusion sensitivity map

Listing D.2: Generating a Multi-Scale Occlusion Sensitivity Map

```python
def get_multi_scale_occlusion_map(img, classifier, n_superpixels, weights):
    """Generates a multi-scale occlusion sensitivity map.

    Combines occlusion heatmaps generated at different superpixel resolutions
    to capture both coarse and fine-grained sensitivities.

    Args:
        img: The input image as a tensor.
        classifier: The classifier.
        n_superpixels: A list of superpixels amount for each scale (coarse to fine).
        weights: A list of weights corresponding to each scale.

    Returns:
        The normalized, refined multi-scale occlusion sensitivity map.
    """

    # Generate occlusion heatmaps at different scales.
    occlusion_heatmap_coarse, coarse_superpixels = get_occlusion_heatmap(img, classifier, n_superpixels[0])
    occlusion_heatmap_fine_1, fine_1_superpixels = get_occlusion_heatmap(img, classifier, n_superpixels[1])
    occlusion_heatmap_fine_2, fine_2_superpixels = get_occlusion_heatmap(img, classifier, n_superpixels[2])

    sensitivity_maps = [occlusion_heatmap_coarse, occlusion_heatmap_fine_1, occlusion_heatmap_fine_2]
    superpixels = [coarse_superpixels, fine_1_superpixels, fine_2_superpixels]

    unique_superpixels = np.unique(coarse_superpixels)
    refined_sensitivity = np.zeros_like(coarse_superpixels, dtype=float)

    # Iterate through each coarse superpixel.
    for sp in unique_superpixels:
        # Create a mask for the current coarse superpixel.
        mask = (coarse_superpixels == sp)

        # Initialize the combined sensitivity.
        sensitivity = 0

        # Iterate through scales (from fine to coarse) and accumulate weighted sensitivities.
        for i, (sensitivity_map, weight) in enumerate(zip(sensitivity_maps[::-1], weights[::-1])):
            # Get the corresponding superpixels at the current scale.
            fine_superpixels = superpixels[::-1][i]  # Accessing scales in reverse order
            fine_mask = (fine_superpixels == sp)  #  Not very efficient - consider improving

            # Calculate and accumulate the weighted average sensitivity if the mask is not empty.
            if np.sum(fine_mask) > 0:
                average_sensitivity = np.mean(sensitivity_map[fine_mask])
                sensitivity += weight * average_sensitivity

        # Assign the combined sensitivity to the refined map.
        refined_sensitivity[mask] = sensitivity

    # Normalize the refined sensitivity map.
    refined_sensitivity = normalize_heatmap(refined_sensitivity)

    return refined_sensitivity
```

Listing D.3: Generating a Superpixel Mask from the Multi-Scale Occlusion Sensitivity Map

```python

def get_superpixels_mask(occlusion_heatmap, r_dimensions=2, n_clusters=5, dataset_name=""):
    """Generates a mask for superpixels based on occlusion sensitivity and clustering.

    Args:
        occlusion_heatmap: The occlusion sensitivity heatmap.
        k_dimensions: The number of clusters to select for the mask.
        n_clusters: The total number of clusters for k-means.
        dataset_name: The name of the dataset (used for mask resizing).

    Returns:
        A tensor mask for use in inpainting.
    """
    # Apply k-means clustering to the occlusion heatmap.
    labels, label_centroid_mapping = apply_k_means(occlusion_heatmap, n_clusters)

    # Extract the cluster labels corresponding to the centroids.
    label_centroid_mapping = [label[0] for label in label_centroid_mapping]
    label_centroid_mapping = np.array(label_centroid_mapping, dtype=np.int32)

    # Create a mask based on whether superpixels belong to the top-r sensitive clusters.
    labels_mask = ~np.isin(labels, label_centroid_mapping[:r_dimensions])
    labels_mask = labels_mask.astype(np.float32)

    # Add contextual cues to the masked region
    labels_mask = replace_zeros_with_ones(labels_mask)

    # Convert the mask to a PyTorch tensor and move it to the GPU.
    mask_res = torch.from_numpy(labels_mask).long().cuda()

    # Resize the mask based on the dataset.
    if dataset_name == "cifar10":
        mask_res = mask_res.unsqueeze(0).unsqueeze(0).expand(1, 3, 32, 32)
    else:
        mask_res = mask_res.unsqueeze(0).unsqueeze(0).expand(1, 3, 256, 256)

    return mask_res
```