Informatics

Evasion Resilience in Variational Bottleneck Injected Deep Neural Networks

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering/Internet Computing

eingereicht von

Patrik Szabó, Bsc
Matrikelnummer 11811341

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ. Prof. Dr. Schahram Dustdar
Mitwirkung: Univ. Ass. Dipl.-Ing. Alireza Furutanpey

Wien, 17. Oktober 2024

Patrik Szabó

Schahram Dustdar

TU WIEN Informatics

Evasion Resilience in Variational Bottleneck Injected Deep Neural Networks

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering/Internet Computing

by

Patrik Szabó, Bsc
Registration Number 11811341

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Schahram Dustdar
Assistance: Univ. Ass. Dipl.-Ing. Alireza Furutanpey

Vienna, October 17, 2024

Patrik Szabó

Schahram Dustdar

Erklärung zur Verfassung der Arbeit

Patrik Szabó, Bsc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 17. Oktober 2024

Patrik Szabó

Danksagung

An erster Stelle möchte ich mich bei meinen Betreuern, Dr. Schahram Dustdar und Dipl.-Ing. Alireza Furutanpey, bedanken. Sie haben mir die Gelegenheit gegeben, diese Arbeit zu realisieren, und die „Distributed Systems Group“ hat nicht nur die erforderlichen Rechenressourcen zur Verfügung gestellt, sondern auch eine angenehme und unterstützende Arbeitsatmosphäre geschaffen.

Mein besonderer Dank gilt Alireza, der mir das Forschungsthema vorgeschlagen hat und mit mir gemeinsam daran gearbeitet hat, es zu konkretisieren. Seine wertvollen und detaillierten Rückmeldungen, die er jederzeit bereitstellte, sowie seine kontinuierliche Unterstützung bei der Umsetzung dieser Arbeit, waren von unschätzbarem Wert.

Zuletzt gilt mein Dank meinen Eltern und meiner Freundin, die mich unermüdlich unterstützt haben. Eure Zuversicht und Liebe haben mir stets Kraft gegeben und waren für mich von unschätzbarem Wert.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisors, Dr. Schahram Dustdar and Dipl.-Ing. Alireza Furutanpey, for giving me the opportunity to work on this project. The "Distributed Systems Group" not only provided the necessary computational resources but also created a pleasant and supportive working environment.

I am especially grateful to Alireza, who proposed the research topic and worked with me to refine and concretize it. His valuable and detailed feedback, which he always provided promptly, as well as his continuous support throughout the realization of this work, have been invaluable.

Lastly, I would like to thank my parents and my girlfriend for their unwavering support. Your confidence and love have always given me strength and have been invaluable to me.

Kurzfassung

Tiefe neuronale Netzwerke (Deep Neural Networks, DNNs) sind in verschiedenen Anwendungen allgegenwärtig geworden, jedoch sind sie äußerst anfällig für adversarielle Angriffe, die ihre Ausgaben leicht manipulieren können. Diese Arbeit untersucht das Potenzial von variationalen Bottleneck-Techniken, insbesondere des Deep Variational Information Bottleneck (DVBI) und des Shallow Variational Bottleneck Injection (SVBI), um die Robustheit von DNNs gegenüber adversariellen Umgehungsangriffen zu erhöhen. Wir führen eine empirische Studie durch, in der Modelle, die mit SVBI, DVBI und traditionellen Architekturen ohne Bottlenecks trainiert wurden, verglichen und ihre Widerstandsfähigkeit gegenüber den neuesten Umgehungsangriffen, einschließlich der FGSM-, EAD-, C&W- und JSMA-Angriffe, analysiert werden. Unsere Forschung bewertet den Einfluss der Platzierung des Bottlenecks auf die Robustheit gegen adversarielle Angriffe und untersucht die Beziehung zwischen Netzwerktiefe und Widerstandsfähigkeit. Die Ergebnisse heben die Wirksamkeit bestimmter variationaler Bottleneck-Strategien zur Verringerung der Anfälligkeit von Modellen gegenüber adversariellen Störungen hervor und bieten Einblicke, wie diese Techniken genutzt werden können, um sicherere KI-Systeme zu entwerfen.

Abstract

Deep neural networks (DNNs) have become ubiquitous in various applications, yet they are highly vulnerable to adversarial attacks that can easily manipulate their output. This paper investigates the potential of variational bottleneck techniques and the role of network depth for adversarial robustness. We conduct an empirical study comparing models trained with Information Bottleneck-based objectives and traditional architectures without bottlenecks. We analyze their resilience against state-of-the-art evasion attacks, including the FGSM, EAD, C&W, and JSMA attacks. Our research evaluates the impact of bottleneck placement on adversarial robustness and explores the relationship between network depth and resilience. Our results highlight the effectiveness of certain variational bottleneck strategies in reducing model vulnerability to adversarial perturbations, providing insights into how these techniques can be leveraged to design more secure AI systems.

Contents

CHAPTER 1

Introduction

1.1 Problem Statement and Motivation

Deep neural networks (DNNs) are becoming increasingly common in everyday applications [41], ranging from image recognition to natural language processing. A common paradigm in modern applications is to offload computationally intensive tasks to remote servers, which incurs significant bandwidth costs. To address this, novel variational techniques have been developed to compress features effectively with lightweight and shallow neural networks without compromising prediction integrity. The Variational Information Bottleneck objective was initially designed to regularize training deep neural networks and directly lends itself to end-to-end train goal-oriented neural compression models [16]. Notably, the IB-based objective was shown to increase adversarial robustness [1]. The basic idea of more lightweight neural feature compression methods is placing a bottleneck at shallow layers and training them with IB-based objectives [10, 11], i.e., they perform *Shallow Variational Bottleneck Injection* (SVBI), However, since their practical implementation differs from the traditional application of the IB principle for compression with Deep Variational Bottleneck Injection (DVBI) [39], it is unclear whether the same assumptions regarding adversarial robustness hold. This thesis aims to resolve the undetermined role of novel paradigms of neural feature compression for adversarial robustness. Specifically, we investigate the relationship between the bottleneck location, network depth, and the effectiveness of the IB objective to mitigate adversarial attacks. We provide a detailed comparative analysis of different DNN models applying SVBI, DVBI, and traditional methods without artificial bottlenecks to evaluate their resilience against a variety of state-of-the-art evasion attacks [12, 6, 5, 29]. Expected outcomes of this research include the successful application of these attacks on selected models, identifying at least one SVBI approach that demonstrates improved resilience, and a comprehensive understanding of how variational bottlenecks can influence the security dynamics of DNNs. Preliminary research suggests that deeper network layers typically extract higher-level semantics of the input data, as the information undergoes multiple transformations throughout the network. Applying a bottleneck at a later stage should allow the network to focus on the most crucial aspects of the data, filtering out potential adversarial manipulations passed on from the shallow layers. Therefore, we assume the overall resiliency is more robust with a deeper bottleneck, as shallow layers generally have more redundancy for a specific task. Nonetheless, it remains to be determined whether the SVBI approach is better than having no bottleneck, especially when compared to split inference [25] approaches that do not apply any IB objective. Through this research, we will provide insights into the protective impacts of bottleneck techniques. We expect the results to contribute to the field by offering empirical evidence and a better theoretical understanding of different defensive mechanisms within neural networks that may yield insights into designing more secure methods.

1.2 Research Questions and Challenges

Through empirical analysis of state-of-the-art attacks on image recognition/classification models and compression schemes, our work will contribute to developing more secure AI systems. To this end, we aim to answer the following research questions:

RQ1: To what extent does the resilience provided by the variational information bottleneck objective against adversarial evasion depend on the layer index?

- To determine whether the redundancy of a minimally informative representation of shallow layers reduces the effectiveness of the IB objective, we apply several adversarial examples against DVBI and SVBI methods.

  Intuitively, the IB objective improves resilience due to regularizing a model to prioritize the most salient features for a given task, i.e., it removes adversarial properties by generally reducing redundancy. However, eliminating redundancy to focus on high-level semantics is an intrinsic property of DNNs. Conversely, shallow layers focus on low-level features and retain high mutual information (MI) with the input [11], i.e., SVBI methods retain more redundancy in the compressed representation.

RQ2: How does the depth of a DNN influence robustness against evasion attacks?

- To determine whether the gap in effectiveness between DVBI and SVBI widens, we evaluate adversarial attacks on related architectures with varying depths (e.g., ResNet-18/50/101).

  Empirical evidence suggests strict inequality in data processing inequality, particularly for discriminative models [38, 11]. In other words, the deeper the layer index, the less redundancy in the representation. Still, it is not apparent whether deeper networks tend to remove information more gradually, such that the MI at the penultimate layer is comparable to a corresponding shallow network (e.g., ResNet-50 and ResNet-18). Alternatively, the penultimate layer of a deeper network may tend to retain less information than a shallower network.

RQ3: To what extent are DNNs with variational bottleneck encoders vulnerable to encoding-specific adversarial attacks?

- To assess the susceptibility of DNNs with variational bottleneck encoders to adversarial attacks that alter encoded representations [43, 8], we will examine how different SVBI and DVBI bottleneck techniques respond to these attacks.

  We will measure the fidelity of encoded outputs under adversarial conditions by quantifying the divergence from intended representations using metrics like Euclidean distance. Furthermore, we will evaluate the impact of these distortions on overall model performance, with a primary focus on classification accuracy, to determine if bottleneck configuration offers better protection against sophisticated attacks targeting the encoding processes.

In the early stages of our work, we identified several challenges we expected to encounter that would need to be addressed to ensure the validity and rigor of our findings. First, inconsistencies in attack success rates present a significant hurdle. To mitigate this, we must standardize attack parameters and ensure a reproducible setup across different models. Additionally, statistical analysis will help account for any variances observed in the results.

Another challenge is the limitation of computational resources. To overcome this, we plan to utilize the university-provided computational equipment to execute our DNN models. It will be crucial to prioritize experiments based on preliminary results, ensuring efficient allocation of these resources to the most promising avenues of research.

Interpreting results can also be challenging, mainly when dealing with the complex behaviors of adversarial attacks. To address this, we will employ visualization tools to analyze the impact of attacks and consult with literature for practical insights, which will help us better understand and explain these complex phenomena.

Finally, the similarity of adversarial approaches may lead to similar final results across different methods. While this is not inherently problematic, as the results will still be usable, it could raise questions about the rigor of our paper. Despite this potential similarity, we must be prepared to justify our methodologies and the significance of our findings.

1.3 Methodology

Our methodology includes:

- **Literature review** to map out the current understanding of DNN vulnerabilities, SVBI and DVBI methodologies, and the evolution of evasion attack mechanisms. Based on extensive preliminary research, four state-of-the-art evasion attacks and one autoencoder attack have been selected, each utilizing a different image generation strategy and attack vector.
  The chosen attack suite includes the following:
  - Gradient-based attacks: FGSM [12], EAD [6], C&W [5], JSMA [29]
  - Attacks on the encoding process itself: Tabacof et al. autoencoder attack [43]
- **Model Training and Adversarial attack creation** involves generating perturbed datasets for our selected evasion attacks (FGSM, EAD, C&W, JSMA). Each attack will be carefully crafted to challenge the DNN models and explore their vulnerabilities. Additionally, we may add random noise or other typical non-adversarial image corruptions [18] (motion blur, noise, etc.) to the input samples to compare a model's general robustness. The adversarial datasets will be used to evaluate the robustness of models under various conditions, focusing on how these perturbations affect the models' accuracy and resilience.
- **Adversarial attack benchmarking** will evaluate adversarial attacks using custom criteria. These will include a binary success rate, which measures whether an attack succeeded at fooling the model or not, presented as a percentage. This straightforward metric allows for an immediate understanding of the efficacy of each attack, and it will be the most essential resulting value. We also include a top-5 accuracy metric to gauge the extent of the potential classification error. The distortion between the original and perturbed images will be measured using pixel-wise error metrics. This data will help determine a boundary of distortion beyond which an image is no longer practically usable, providing a clear metric for assessing the trade-off between image fidelity and vulnerability to attacks. Where applicable, we will also use the Bits Per Pixel (BPP) metric to measure the information retained through the compression process in a bottleneck-injected model. Following this, statistical and theoretical analyses will be employed to interpret the results, seeking correlations between bottleneck/compression techniques and improvements in model resilience.

We evaluate the experiment results of selected DNN models under the selected attack scenarios. To this end, we will devise a combination of datasets and models best suited for their classification on the testing rig. The susceptibility of these underlying classification models to evasion attacks of many kinds is well documented in the literature. The specific goal of this thesis is to examine how the robustness of these models might be improved thanks to bottleneck injection. Therefore, the choice of the underlying model itself is of lesser importance. Finally, we will generate and apply a suite of adversarial examples based on the specifications detailed in the corresponding papers from the literature review.

1.4 Structure

The rest of this paper is structured as follows. Chapters 2 and 3 include a detailed summary of the state-of-the-art field of deep neural networks, Information Bottlenecks, and evasion attacks that are relevant to our research. It outlines basic terms and definitions that are referenced throughout this work. Chapter 4 describes our experimental methodology, including the design and evaluation metrics. Chapter 5 consists of the actual experimental part of this thesis, in which we run our experiments and empirically analyze and evaluate the information gathered. Chapters 6 and 7 include a discussion and interpretation of our findings and the summary and contributions of this work, respectively.

CHAPTER 2

Background

This chapter provides the necessary background for understanding the core concepts relevant to this thesis. It covers the fundamentals of deep neural networks, their vulnerabilities to adversarial attacks, and key defense mechanisms. Additionally, it introduces the Variational Information Bottleneck (VIB) technique, which, as we posit, might help enhance DNN robustness. These topics form the foundation for the subsequent exploration of DNN security and the strategies to improve resilience against evasion attacks.

2.1 Deep Neural Networks

Deep neural networks are a key component of modern artificial intelligence, enabling progress in computer vision, natural language processing, and autonomous systems [27]. They consist of multiple layers of neurons, each transforming the input data into increasingly abstract representations, allowing for learning complex patterns and generating accurate predictions. Deep learning involves training DNNs on large datasets, allowing them to model the underlying data distributions. This is done by optimizing a loss function, quantifying the difference between predicted and actual outputs. Techniques such as backpropagation and gradient descent are employed to adjust network weights and minimize this loss, thereby improving performance over time.

While machine learning encompasses many models, including classical approaches like decision trees, support vector machines, and random forests, our focus is specifically on deep neural networks within deep learning. Unlike traditional machine learning models, which rely on handcrafted features and typically involve fewer layers, DNNs consist of multiple layers that progressively extract complex hierarchical representations from data. This depth enables DNNs to capture intricate patterns, making them well-suited for high-dimensional data and tasks such as image classification and natural language processing.

A key feature of DNNs is their ability to generalize from training data to unseen data. However, this generalization also makes DNNs vulnerable to evasion attacks, where small, deliberate perturbations in input data cause incorrect predictions. These attacks reveal fundamental weaknesses in DNNs, underscoring the need for robust defense mechanisms.

One approach to enhancing DNN robustness is the bottleneck principle [45]. This method compresses the information within the network by using a bottleneck layer (Figure 2.1), forcing the model to retain only the most relevant features for a task. The Variational Information Bottleneck technique [1], an extension of this principle, further improves compression by employing variational inference. The VIB method balances between retaining essential information for task performance and minimizing redundant information that adversarial attacks could exploit. By compressing feature representations, VIB can improve resilience to input perturbations, enhancing both security and generalization.

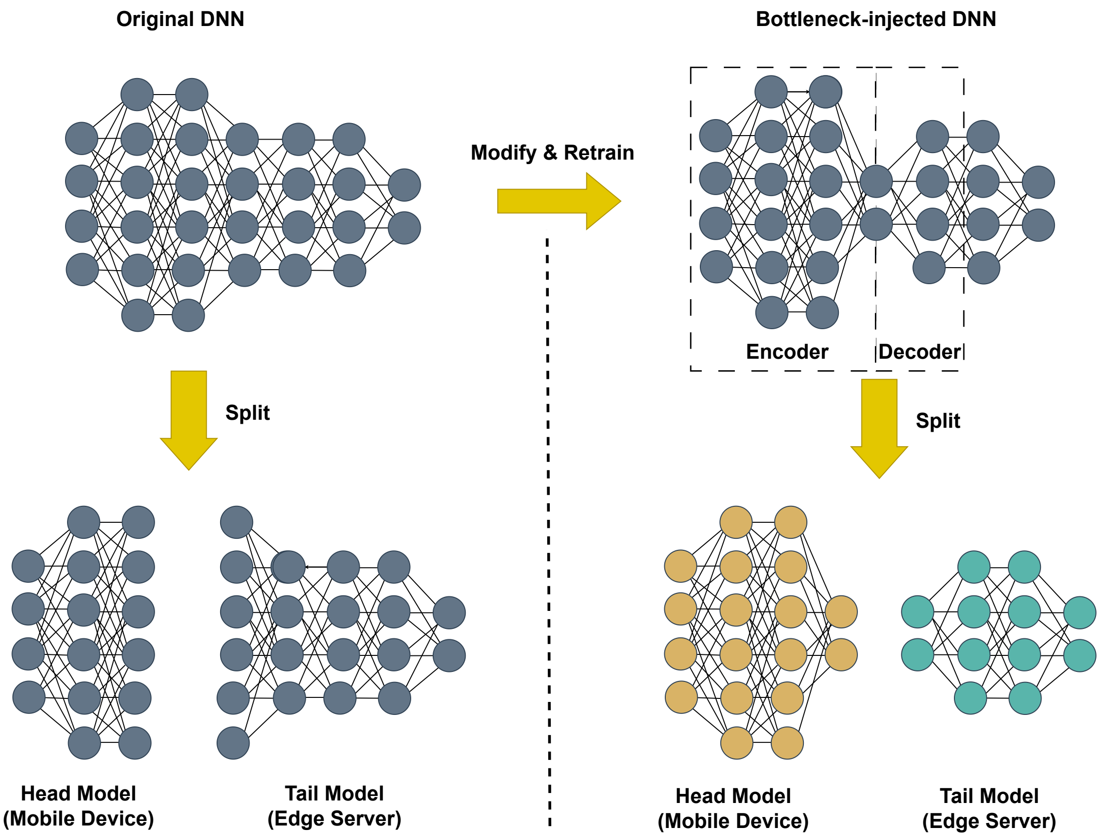


Figure 2.1: Visualization [25] of an injected bottleneck as it is used in split computing to unburden an edge computing device on the right, compared to a typical DNN on the left.

2.2 Exploitability of DNNs

The widespread use of DNNs has exposed significant security vulnerabilities, particularly to adversarial attacks [3]. These attacks involve subtle perturbations to input data that result in incorrect outputs. Understanding these vulnerabilities is critical for securing AI systems.

Adversarial attacks exploit the non-linear and high-dimensional nature of DNNs. Goodfellow et al. [12] showed that DNNs are susceptible to adversarial examples, where inputs are intentionally modified to induce errors. The Fast Gradient Sign Method (FGSM) calculates the gradient of the loss function concerning the input, adjusting the input to maximize the loss and create adversarial examples. These vulnerabilities pose significant risks, particularly in safety-critical applications. For example, Gu et al.[15] demonstrated how small changes could cause a DNN to misclassify a stop sign as a speed limit sign, posing risks for autonomous vehicles. In other fields, such as medical imaging, adversarial attacks could result in misdiagnoses, while in financial systems, they could enable fraudulent transactions.

The susceptibility of DNNs to adversarial attacks is often attributed to the sensitivity of their latent spaces to small perturbations. In high-dimensional input spaces, data points from different classes are separated by complex, non-linear decision boundaries. Adversarial perturbations can exploit these boundaries by introducing minimal changes to the input sufficient to cross into a different class region, leading to misclassification [36]. This phenomenon is exacerbated by the fact that DNNs may not generalize well to inputs that deviate slightly from the training data distribution, making them vulnerable to carefully crafted adversarial examples.

This phenomenon can be understood by examining how DNNs process inputs through successive layers. Each layer transforms the input into a new representation in the latent space, where ideally, data points belonging to the same class become more tightly clustered, and those from different classes become increasingly separated. However, in practice, these transformations can amplify the effect of small perturbations. Perturbed inputs close to the boundary in the input space may traverse disproportionately large distances in the latent space, leading to significant shifts in their classification.

Several defense strategies have been proposed to counter adversarial attacks. These will be explored further in Chapter 3. Despite progress, defenses often lag behind new attack techniques, highlighting the need for continuous research into the security of DNNs. DNN vulnerabilities also extend to privacy concerns. Shokri et al. [37] showed that DNNs are vulnerable to membership inference attacks, where an attacker can determine whether a particular data point was part of the training set. This has significant privacy implications, especially with sensitive data.

2.3 Adversarial Attacks on DNNs

Adversarial attacks represent a significant challenge for deploying AI systems. They are classified into white-box and black-box attacks. White-box attacks assume complete model knowledge, including architecture and gradients, allowing for highly effective adversarial examples. In contrast, black-box attacks assume no access to model details and are generally more challenging but more realistic for real-world scenarios.

The Fast Gradient Sign Method (FGSM) [12] is a well-known adversarial attack method, visible in Figure 2.2, which will be explored in greater detail in Chapter 3. Building on FGSM, Kurakin et al. proposed the Basic Iterative Method (BIM) [22], which applies FGSM iteratively with smaller step sizes. This iterative approach allows for finer control over the perturbation and can generate more potent adversarial examples. Further relevant evasion attack methods will be examined in the Related Work Chapter of this paper.

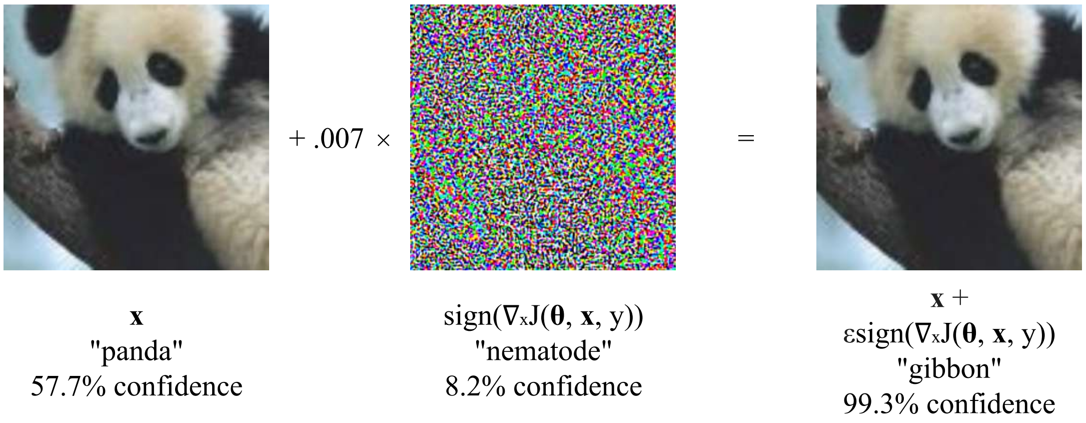


Figure 2.2: Example of the FGSM attack applied against GoogLeNet [12]. By introducing a barely noticeable vector, where each element matches the sign of the gradient of the cost function concerning the input, GoogLeNet's image classification can be altered.

In addition to these gradient-based attacks, evolutionary algorithms have been employed to generate adversarial examples. Alzantot et al. introduced [2] a black-box attack method that uses a genetic algorithm to evolve adversarial examples iteratively. This approach does not require gradient information and can effectively generate adversarial examples by querying the target model.

Defending against adversarial attacks is an active area of research. Adversarial training, where the model is trained on a mix of clean and adversarial examples, has shown promise. However, this approach is computationally intensive and may not generalize to unseen attacks [23].

2.4 Perceptible vs. Imperceptible Noise

In the context of digital imaging and image classification tasks, particularly when examining the robustness of DNN classifiers under adversarial attacks, the distinction between perceptible and imperceptible noise becomes crucial, especially when considering the quantization of pixel values. For instance, in an 8-bit per pixel image, each pixel can take on one of 256 discrete intensity values ranging from 0 to 255. Perceptible noise refers to perturbations significant enough to alter the quantized pixel intensity values. In other words, if the noise introduced by an adversarial attack, such as those used in our experiments, is sufficient to shift the pixel value from one quantized level to another (e.g., from 100 to 101), it is considered perceptible. This perceptible noise can lead to visible changes in the image, which might not only degrade the visual quality but also impact the performance of DNN classifiers [13].

On the other hand, imperceptible noise refers to perturbations that are too small to affect the quantized pixel values. In an 8-bit image, the noise is smaller than the smallest step between quantized levels (i.e., smaller than one unit of intensity). Even if the pixel intensity is altered by a small amount (e.g., from 100 to 100.4), the quantization process would round this value back to 100, meaning the perturbation is not reflected in the displayed image. This type of noise is imperceptible in pixel intensity and does not result in visible changes. However, it may still significantly affect the classifier's performance due to the sensitivity of DNNs to even small perturbations in input data.

This distinction between perceptible and imperceptible noise is crucial in evaluating the vulnerability of classifiers to adversarial attacks. Attack methods that generate imperceptible noise often succeed in inducing misclassifications without being visually detectable [19], creating security challenges in classification models.

2.5 Variational Information Bottleneck

Compression methods based on the Information Bottleneck (IB) principle are designed to minimize the bitrate while preserving task-relevant information. These methods balance retaining predictive performance and removing redundancy, as formalized by Tishby et al. [44, 46]. Recent works, such as those by Singh et al. [39] and Dubois et al. [9], have demonstrated that by focusing on task-centric compression objectives, it is possible to achieve substantial rate savings without compromising downstream task performance. This is achieved by optimizing a trade-off between the mutual information of the input and the latent representation (I(X;Z)) and the mutual information of the latent representation and the output (I(Z;Y)). The IB objective can be formulated as:

$$\min_{p(z|x)} I(X;Z) - \beta I(Z;Y)$$

Where β is a Lagrange multiplier that controls the trade-off between compression and predictive power.

Alemi et al. extended [1] the IB principle to deep learning by introducing the Variational Information Bottleneck. They proposed a variational approximation to the IB objective, leveraging variational inference techniques to make the optimization tractable for high-dimensional data. The objective function of the VIB method is given by:

$$J_{IB} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[-\log q(y_n | f(x_n, \epsilon)) \right] + \beta KL \left[p(Z|x_n), r(Z) \right] \tag{2.1}$$

The Variational Information Bottleneck approach has been demonstrated to improve the generalization and robustness of deep neural networks. By imposing a bottleneck that limits the information passed through the network, VIB forces the model to focus on the most relevant features, reducing overfitting and enhancing its resilience to noisy and adversarial inputs. For example, models using VIB have shown greater robustness to adversarial attacks than standard neural networks, as the compressed representation complicates the task of causing misclassifications via adversarial perturbations.

Recent advancements have extended the applicability of VIB techniques. For example, VIB has been combined with generative models to develop more powerful representations. Kingma et al. [21] introduced the Variational Autoencoder (VAE), which employs a similar variational approach to learn latent representations for generative tasks. Integrating VIB and VAE frameworks facilitates the compression and generation of high-dimensional data, supporting unsupervised learning and generative modeling applications.

However, VIB presents certain challenges. A significant issue is balancing the trade-off between compression and predictive performance. The choice of β is critical, as excessive compression may result in losing important information, while insufficient compression fails to achieve the desired robustness and generalization. Furthermore, the variational approximation introduces additional complexity in the training process, requiring careful tuning of variational parameters and optimization techniques.

2.6 Deep Variational Information Bottleneck

Building on the VIB framework, Singh et al. [39] formulated a rate-distortion optimization problem that refines the trade-off between compressibility and task-specific performance. The optimization problem is defined as:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} \sum_{x,y \in D} L(y, \hat{y}) + \lambda \cdot -\log_2 p(\hat{z}; \phi)$$

Here, $L(y, \hat{y})$ represents the distortion term, corresponding to the prediction error, while the term $-\log_2 p(\hat{z}; \phi)$ accounts for the bit rate, representing the compressibility of the latent representation $\hat{z}$. The trade-off parameter λ adjusts the balance between these two objectives. This formulation enables the model to produce highly compressible feature representations while maintaining high accuracy, which is crucial for tasks such as image classification, where storage and transmission of features are significant concerns.

2.7 Shallow Variational Bottleneck Injection

SVBI is a neural feature compression method aimed at enhancing the efficiency and robustness of deep neural networks by targeting the shallow representation of foundational models for rate-distortion optimization, enabling task-agnostic compression without discarding critical information for downstream tasks [11, 10, 26].

By leveraging the high mutual information between shallow representations and the input, SVBI retains essential task-relevant features with minimal information removal, making it more generalizable to a broader range of tasks than deeper approaches. However, the minimal removal of information relevant to the task suggests that this approach may be less resistant to input perturbations, including adversarial attacks. SVBI reduces data size while maintaining performance, making it suitable for bandwidth-constrained environments such as satellite computing and edge devices. Experiments have demonstrated significant bitrate reductions without sacrificing accuracy, especially in tasks like image classification and object detection. Additionally, the method is easily generalizable across different neural network architectures with minimal adjustments required for integration into pre-trained models. However, a key challenge lies in balancing compression and reconstruction quality, as improper tuning may lead to the loss of critical information during compression.

SVBI trains neural codecs by replacing the distortion term in the rate-distortion objective of variational image compression models with head distillation (HD). In the Knowledge Distillation (KD) framework, the codec acts as the student, learning from the shallow layers of a foundational model, or the teacher, as shown in Figure 2.3. Unlike traditional KD, HD takes a different approach, focusing on reconstructing the representation of the foundational model. The intuition behind SVBI is that if a codec can successfully reconstruct this representation, it will be sufficient for all tasks associated with the model.

The uses of this approach have been thoroughly examined in the literature. For example, the Entropic Student [26] leverages knowledge distillation and neural image compression to compress intermediate feature representations efficiently. This approach involves teacher and student models with a stochastic bottleneck and a learnable prior for entropy coding. The student model is trained to match the teacher's intermediate features, and the bottlenecked features are compressed and transmitted to the edge server, where the bulk of the computation is completed.

FrankenSplit [10] introduces a general framework for SVBI and demonstrates its efficacy for latency-critical and performance-critical visual applications in edge-cloud computing settings where the sender has only limited computational resources. FOOL [11] extends this approach for Orbital Edge Computing (OEC). This method effectively partitions high-resolution satellite imagery to maximize throughput, embedding contextual information

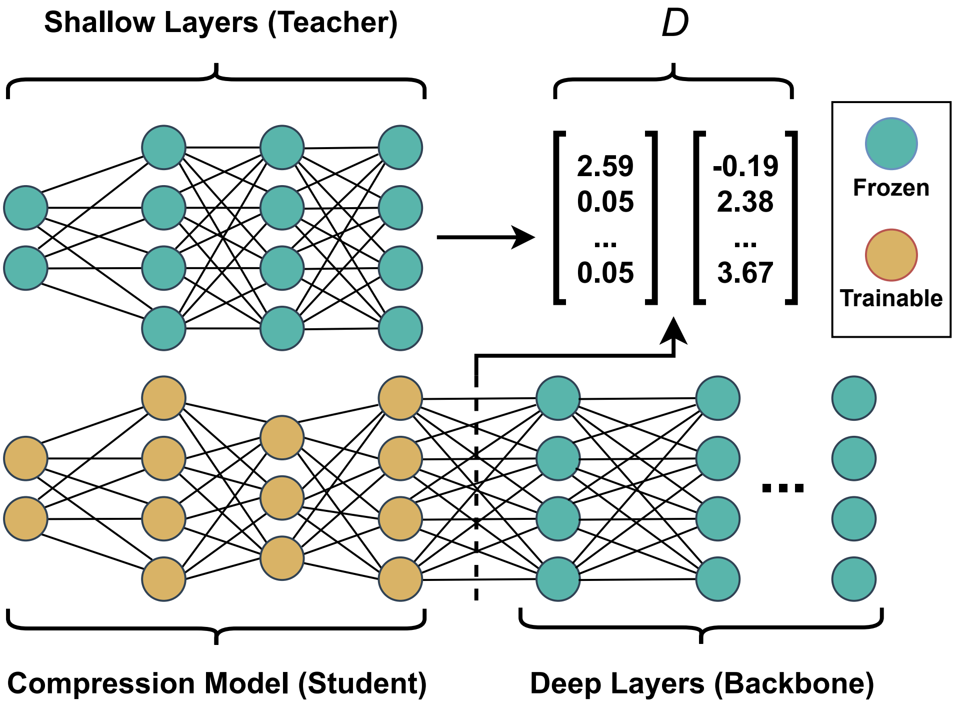


Figure 2.3: Head distillation visualization. In this example, teacher penalizes the student model for an insufficient approximation of the shallow representation of a pre-trained feature extractor based on distortion loss.

and leveraging inter-tile dependencies to reduce transfer costs with negligible overhead. Remarkably, FOOL maintains high prediction performance while allowing for a significant increase in downlinkable data volume without requiring prior knowledge of downstream tasks. By targeting shallow representations, FOOL ensures that critical information necessary for a wide range of prediction tasks is preserved, even under varying and unpredictable conditions.

CHAPTER 3

Related Work

This section reviews notable studies on DNNs' vulnerability to adversarial attacks, focusing on methods directly related to crafting adversarial examples and examining and improving robustness.

3.1 Adversarial Example Creation

The susceptibility of DNNs to adversarial examples was first highlighted by Szegedy et al. [42], who demonstrated that small, imperceptible perturbations to input data could lead to significant misclassifications. Building on this, Goodfellow et al. [12] introduced the Fast Gradient Sign Method, a white-box attack that efficiently generates adversarial examples by leveraging the gradient of the loss function. FGSM works by adjusting the input along the gradient's direction, with the perturbation defined as:

$$x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y))$$

where x is the input, ϵ controls perturbation magnitude, and $\nabla_x J(x, y)$ is the gradient of the loss concerning the input. FGSM laid the groundwork for many subsequent attacks, and it will be employed in this thesis as a baseline to evaluate the robustness of bottleneck-injected DNNs.

Building on FGSM, Carlini et al. [5] proposed the Carlini & Wagner (C&W) attack, which has become one of the most potent gradient-based attacks. The C&W attack is formulated as an optimization problem that balances the perturbation's magnitude and the degree of misclassification across different L_p norms (L0, L2, Linf). The L2 attack, for instance, minimizes:

$$\min \|x' - x\|_p + c \cdot f(x')$$

where x' is the adversarial example, and c balances perturbation size and misclassification confidence. This attack will be used to stress-test bottleneck-injected models due to its ability to generate high-confidence adversarial examples, providing a rigorous evaluation of model defenses.

In addition to L2 and Linf norm-based attacks, EAD (Elastic-net Attacks to DNNs) [6] introduces L1-oriented adversarial examples, adding an L1 term to the C&W objective. This method is beneficial for producing sparse perturbations, which can fool DNNs while maintaining minimal changes to the input. EAD's dual-norm optimization will be an alternative benchmark for evaluating how variational bottleneck injection handles diverse attack strategies.

Another distinctive approach is the Jacobian-based Saliency Map Attack (JSMA) [29], which constructs adversarial examples by identifying and perturbing input features most critical to the classifier's decision-making process. Unlike gradient-based methods, JSMA uses forward derivatives to create a saliency map, guiding perturbations to specific input features. Given that variational bottleneck techniques may alter feature representations, testing JSMA will allow us to explore how bottleneck injection influences feature saliency and adversarial resilience.

The adversarial example generation techniques reviewed here, particularly FGSM, C&W, EAD, and JSMA, are central to this thesis's experimental evaluation of bottleneck-injected models. These attacks will assess whether variational bottleneck techniques can enhance robustness by limiting the network's susceptibility to adversarial perturbations.

3.2 Improving Robustness of DNNs

Improving the robustness of DNNs is essential for enhancing their resilience to adversarial attacks and input perturbations. In this context, robustness refers to a model's ability to maintain performance despite such interference. Various defense strategies have been studied, and this section focuses on those relevant to bottleneck injection.

Adversarial training, introduced by Goodfellow et al. [12], retrains networks on a mix of clean and adversarial examples, pushing the model to learn more robust decision boundaries. While effective, it is computationally expensive and often reduces accuracy on clean data, motivating the exploration of alternative methods such as bottleneck injection, which aims to improve robustness without this trade-off.

Defensive distillation, proposed by Papernot et al. [30], trains a secondary model on soft class probabilities to smooth decision boundaries. Although initially promising, Carlini et al. [4] showed that distilled models remain vulnerable to advanced attacks like the C&W attack, reconsidering distillation as a defense.

Input preprocessing techniques, such as JPEG compression and bit-depth reduction [17], have been explored to reduce adversarial noise. However, these methods often degrade performance on clean data, limiting their practical use. Bottleneck injection shares the goal of filtering noise but does so within the network architecture itself.

Architectural changes have also shown potential, such as adding noise layers or altering activation functions. Xie et al. [47] introduced feature denoising by adding non-local blocks to remove noise from intermediate features. This strategy improves robustness by addressing noise at multiple levels in the network. In contrast, bottleneck injection methods like the Variational Information Bottleneck [1] focus on compressing and regularizing information flow through the network, compelling the model to retain only the most relevant features. This compression may naturally enhance robustness by reducing the network's sensitivity to irrelevant perturbations.

In this thesis, we explore variational bottleneck injection as a mechanism for improving DNN robustness. By introducing a bottleneck layer that compresses intermediate representations, VIB aims to mitigate the impact of adversarial perturbations while maintaining high task performance. This method contrasts with the more computationally expensive techniques like adversarial training and preprocessing, offering a potentially more efficient and elegant solution. To our knowledge, the use of this method for defensive purposes against adversarial image perturbations has so far been unexplored in literature.

3.3 Targeting Autoencoders

The selected adversarial example types assess the robustness of SVBI and DVBI models against state-of-the-art DNN classifier attacks compared to unsecured models. However, introducing bottlenecks may create additional attack vectors. Therefore, we examine the vulnerability of the autoencoder layer inserted before the classifier.

Chen and Ma's Fast Threshold-constrained Distortion Attack (FTDA) [7] generates adversarial examples for neural image compression (NIC) models. FTDA introduces minor perturbations to an image, significantly distorting its decompressed version when processed by a NIC model. Like other adversarial techniques targeting classifiers, the process involves adjusting noise to balance detectability and distortion maximization. The noise is refined to maximize the difference between the original and adversarial decompressed images. While the approach focuses on the perturbed image input for classification, our interest lies in the internal, compressed representation passed directly to the classifier.

Tabacof et al. [43] also use adversarial perturbations to target autoencoders, aiming not only to disrupt reconstruction but also to induce the encoder to produce a completely different target image. This would undermine the potential defensive role of autoencoders in de-noising classifier inputs. The study shows that while autoencoders offer some protection, small perturbations can still succeed. We hypothesize that if perturbations alter the internal representation to resemble a different image, the classifier may not need to be fooled directly. The attack generates noise to match the internal representation of a target image and applies it to an input. Trying to generate noise to match the target image directly fails.

It is difficult to quantitatively evaluate the approach's effectiveness, as the result is not a label but a decompressed image. We should not encounter such issues because we do not have to reconstruct the compressed image into a human-readable representation.

CHAPTER 4

Experimental Setup

This chapter details the experimental setup and execution methodology used to investigate the resilience of deep neural networks against adversarial attacks by implementing variational bottleneck techniques. The structure and processes described here are designed to ensure replicability, accuracy, and comprehensive evaluation of our hypotheses and research questions. To properly compare and evaluate our selected classification models, datasets, and evasion attacks, we have set up an experimental pipeline to parse inputs to a standard format, train the models chosen on selected datasets, generate perturbed images, and produce statistics and norms for later analysis. In the following sections, we describe the architecture of the pipeline in greater detail, explaining our process to ensure reproducibility. Our data collection and preparation processes will also be explored in this chapter, as well as the metrics we have chosen to empirically and objectively evaluate our results, including the reasoning behind our choices.

4.1 Experiment Design

Our experimental design leverages Python's extensive libraries and tools, ensuring an efficient and effective setup. We use Torchvision [24] for dataset handling and model architectures and CUDA to utilize GPU acceleration, significantly speeding up the training and evaluation processes.

The experimental setup is configured with high-performance components to ensure optimal performance and reliability for deep learning tasks. The system is equipped with two Micron 32GB DDR4-3200 ECC UDIMM 2Rx8 C $L_2$2 memory modules, providing a total of 64GB of error-correcting code (ECC) memory, which is essential for maintaining data integrity during intensive computations. At the core of the setup is an AMD Ryzen 7 5700G processor featuring eight cores and 16 threads, supported by a single GeForce RTX 4070 Ti graphics card to handle parallel processing demands. While this combination provides adequate speed and stability for most tasks, our computational resources are limited. As a result, we focused primarily on lower-dimensional images and smaller datasets to ensure feasible experimentation. This configuration allowed us to efficiently explore different training and attack parameters within the constraints of our resources, providing thorough evaluations.

4.1.1 Dataset Selection

When selecting datasets for our experiments, we focused on balancing complexity, size, and relevance to the task. Our goal was to include datasets that are diverse in structure, manageable within our computational constraints, and capable of highlighting the effects of adversarial perturbations on classification tasks. We prioritized datasets that provide a range of challenges, from standard benchmarks to more complex, high-resolution image sets.

SVHN was chosen for its complexity in digit recognition tasks, while MNIST is a standard benchmark for handwritten digit classification, consisting of ten classes. We opted for the cropped digits format of SVHN, which contains 73,257 training images and 26,032 test images, while the MNIST dataset consists of 60,000 training images and 10,000 test images. CIFAR-10 was selected due to its manageable size and everyday use in image classification tasks, with a train/test split of 50,000/10,000. Initial experiments were done using CIFAR-100. However, because of the relatively low number of training images available for each class, we have failed to train our classification models to be accurate enough to see the apparent adverse effects of the perturbations. As the name would suggest, CIFAR-10 contains ten classes. ImageNet64, a downscaled version of ImageNet, is used to reduce training time while maintaining task complexity, and Felidae, a subset of ImageNet including high-resolution(224x224) images of cats, provides a comparison point for high-resolution image classification like the entire ImageNet dataset. The ImageNet64 dataset comprises 1,281,167 training data and 50,000 test data with 1,000 classes. Felidae, being a subset, is smaller, with only 9,100 training images and 350 test images, sorted into seven classes.

4.1.2 Models and Training

Our experimental model selection consists of ResNet variants: ResNet-18, ResNet-50, and ResNet-101. These models are chosen for their differing depths and complexity. ResNet-18, being lightweight, is suitable for initial experiments and baseline comparisons. ResNet-50, with increased depth, provides insights into the impact of intermediate depth on performance and robustness. ResNet-101, the deepest model in our selection, allows us to explore the effects of extensive depth on adversarial robustness and model performance. Our model-training pipeline is displayed in Figure 4.1. We used *torchvision.models* TorchModels [34] to load the "hollow" models that we trained from scratch.

The training process involves using a standardized approach. Cross-entropy loss is employed as the loss function for classification tasks. We use a combination of Stochastic Gradient Descent (SGD) with momentum and weight decay and the Adam optimizer to

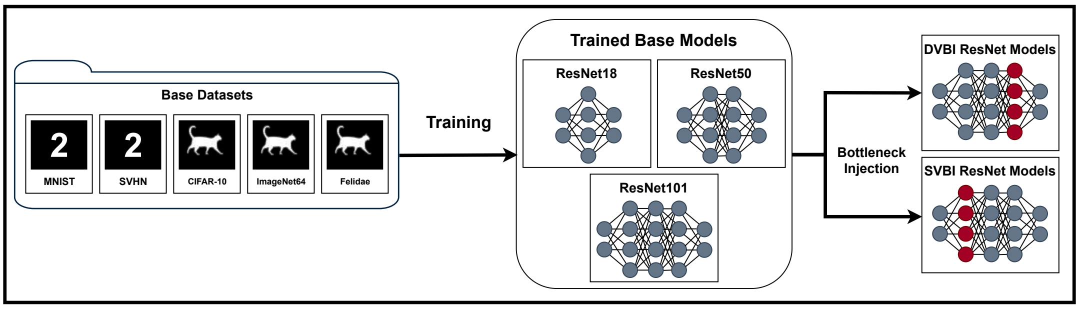


Figure 4.1: A visualization of our model training process.

enhance convergence and generalization based on what worked best during our initial experiments with the smaller and faster ResNet-18 model. A learning rate scheduler, like ReduceLROnPlateau, adjusts the learning rate based on epoch loss, ensuring optimal training dynamics. Gradient clipping is applied to prevent exploding gradients and ensure stable training. We also employ an early exit strategy, which stops the learning process to combat diminishing returns. Because of the computational- and time cost of using cross-validation (CV) during training, specifically in Deep Learning with large datasets, we have decided not to utilize CV for our purposes. We do not use any existing validation datasets, and we utilize whole training sets for training with no validation split.

4.1.3 Image Perturbations

Adversarial attacks are generated using the torchattacks [20] library, which provides many state-of-the-art attack methods out-of-the-box. We selected FGSM, EAD, C&W, and JSMA for their distinct characteristics and perturbation methods. FGSM is a well-known method for adversarial example generation, which we have covered in the previous sections. It was chosen for its simplicity and effectiveness in generating adversarial examples. EAD was selected for its ability to craft adversarial examples using an elastic-net regularization that combines L1 and L_2 norms. The C&W attack, also extensively explored in Chapters 2 and 3, is known for its strong performance in creating high-confidence adversarial examples, making it an essential part of our evaluation. JSMA focuses on perturbing specific input features, which we thought would provide a different perspective on the model's vulnerability. This attack tends to be memory-inefficient when applied to larger images, like the Felidae dataset. For this reason, we have implemented an adjusted version of the JSMA attack called the JSMAOnePixel, inspired by Kenny [40]. Compared to the standard JSMA attack, the OnePixel variant identifies only one pixel with the highest impact on the classification and changes only it each iteration. Supposedly [40], this small change significantly improves the processing time while keeping the effectiveness essentially the same as JSMA. As can be observed from Figure 4.2, the final perturbed images look indistinguishable between base JSMA and JSMAOnePixel. We use the JSMAOnePixel variant on the Felidae and ImageNet64 datasets while using regular JSMA for all other datasets.

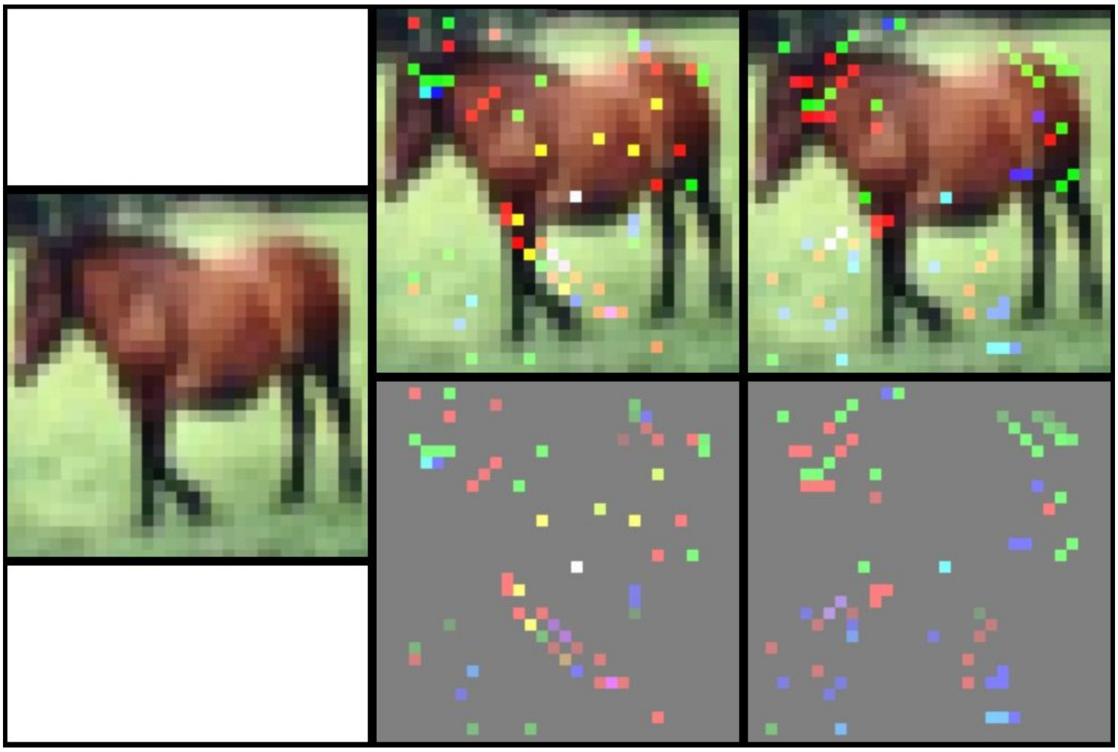


Figure 4.2: An example of a CIFAR-10 image (left) perturbed with JSMA (middle) and JSMAOnePixel (right). Base JSMA does not scale well beyond CIFAR-sized inputs.

In addition to these standard evasion attack methods developed to fool image classifiers specifically, we also aim to examine how specific ResNet models with specific bottlenecks deal with images altered with common image corruptions. We then compare the classification accuracies with the base models' accuracy. We do this, on the one hand, to test the overall ability of the model to deal with altered inputs that have not been adversarially perturbed specifically with the goal of evasion and also to see if bottleneck-injected models deal better with these more straightforward, more common corruptions. This should put into perspective the deterioration in accuracy with evasively perturbed images. The selected alterations we examine are Gaussian noise, defocus blur, motion blur, and low contrast. We considered analyzing the pixelation corruption as well, but since most of our datasets are relatively low resolution, the effects would either not be noticeable, or the inputs would get too heavily distorted to the point where fair comparisons could not be drawn. Our selection draws from the image corruptions presented by Hendrycks et al. [18]. Figure 4.3 shows the complete testset collection we prepare for our experiments.

4.1.4 Deep Variational Bottleneck Injection

Once base models are trained and evaluated, we repeat the process with deep bottleneck-injected versions of ResNet-18, ResNet-50, and ResNet-101, crafted based on the work

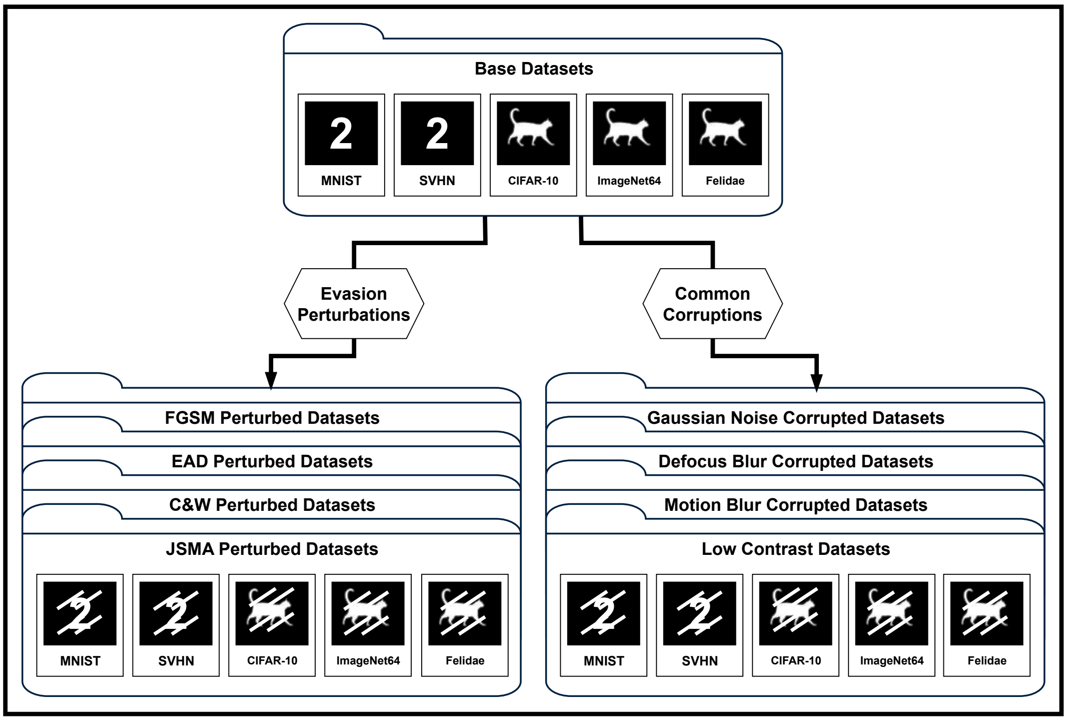


Figure 4.3: A visualization of the results of our dataset preparation process.

done by Singh et al. [39]. We build upon the ResNet architectures by incorporating an entropy bottleneck layer, which encourages the network to produce latent representations that are more compressible while maintaining high classification accuracy. The entropy bottleneck layer forces the intermediate representation to be compressible by minimizing its entropy. The network has three main components: feature extraction, entropy bottleneck, and classification. The modified architecture can be summarized as follows:

- Feature Extraction: The input image is passed through the feature extraction layers of the ResNet. This part of the network includes all convolutional and pooling layers up to the penultimate layer of the original ResNet.
- Entropy Bottleneck: The output of the feature extraction layers is passed through an entropy bottleneck. This layer quantizes the feature maps and computes the likelihoods of the quantized values. The entropy bottleneck is designed to produce a latent representation optimized for compressibility.
- Classification: The quantized latent representation is then passed through the final fully connected layer to produce the class scores.

By incorporating the new layer, we effectively create a bottleneck that serves two purposes: reducing the dimensionality of the feature maps and encouraging a distribution that is more amenable to compression. The experimental pipeline beyond this point remains the same as with the baseline models. The training parameters for some models had to be adjusted after implementing the bottleneck layer for the DVBI model to reach an accuracy comparable to the base model accuracy. A detailed summary of training parameters and methods can be found in Chapter 5, where we also present our findings regarding the accuracy and potential robustness gains or losses that the deep bottleneck layer provides.

4.1.5 Shallow Variational Bottleneck Injection

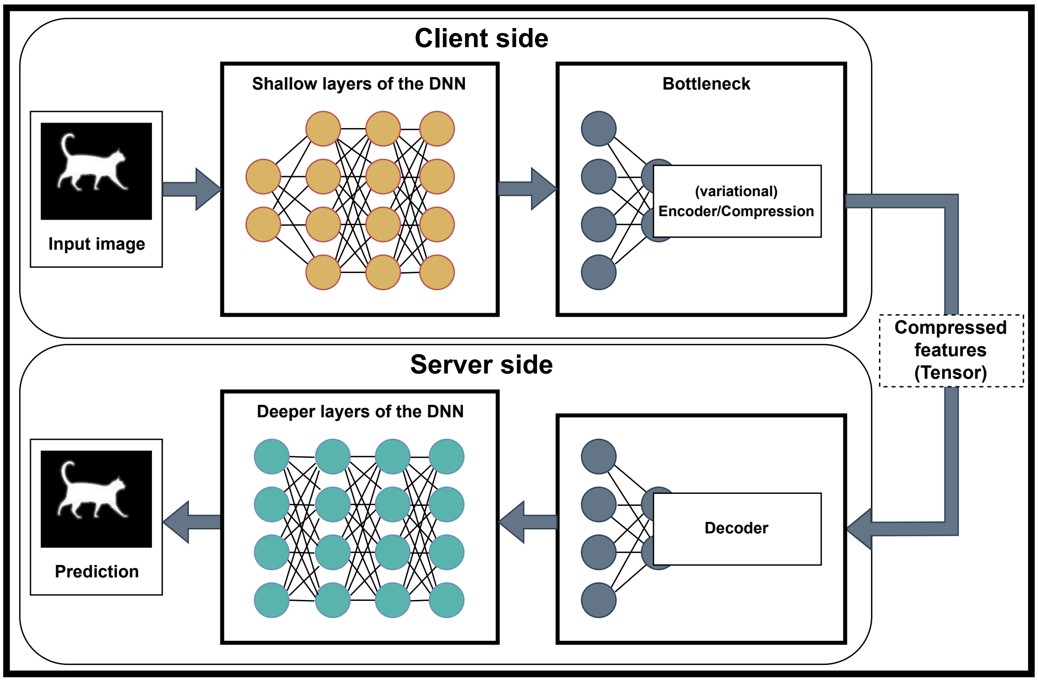

Client side
Shallow layers of the DNN
Bottleneck
Input image
(variational) Encoder/Compression
Compressed features (Tensor)
Server side
Deeper layers of the DNN
Prediction
Decoder

Figure 4.4: SVBI scenario motivated by the need to compress high-dimensional features early to reduce the volume of data that needs to be transmitted to a server while maintaining task performance.

For our experiments, we implemented the shallow variational bottleneck as described in [10]. The bottleneck injection was applied to the respective pre-trained models, generated as described at the start of this chapter. It is worth noting that with standard SVBI, training does not need to be performed separately for each dataset. Typically, the model can be trained once on a large and diverse dataset, such as ImageNet, and it will generalize effectively to other datasets like CIFAR or SVHN. The dataset does not have to be ImageNet specifically but should be sufficiently large and diverse to capture a broad range of features. However, in our case, we train SVBI separately for each dataset to maintain consistency with our baseline, which was trained individually for each dataset.

Following this, the deeper layers or the prediction head can be fine-tuned as usual. Figure 4.4 outlines the general setup of an SVBI model used in a split computing scenario.

4.1.6 Attacking the Bottleneck

Next, we explore the application of the Tabacof [43] attack to perturb the MNIST dataset to deceive a variational information bottleneck model. The primary objective of this attack is to manipulate the input images such that the bottleneck layer interprets them as different, targeted images. This experiment aims to evaluate the robustness of the VIB model against adversarial perturbations and understand the impact of these perturbations on the latent representations. We want to examine our theory that bottleneck-injected models would be more susceptible to such attacks than the base models, as the attack targets the injected bottleneck layer.

The Tabacof attack, proposed by Tabacof et al., targets the latent space of variational autoencoders.The attack generates adversarial examples by maximizing the Kullback-Leibler (KL) divergence between the original and target latent distributions. This results in the model interpreting the perturbed input as the target image. The core idea of the Tabacof attack is to add a carefully calculated perturbation to the input image, such that the resulting latent representation is pushed towards the latent representation of a different, specified target image. This is achieved by optimizing the adversarial noise through gradient-based methods, explicitly targeting the KL divergence in the latent space.

As described by Tabacof et al. [43], the attack was designed and tested with the SVHN and MNIST datasets. We have decided to limit ourselves to the MNIST dataset for our experiments. We implemented the code published by the authors [32] into our pipeline, adapting it for our purposes and to work with our pre-existing setup. To implement the Tabacof attack, we followed these steps:

1. **Model Preparation**:
    - We utilized a ResNet-based VIB model pre-trained on the MNIST dataset. The model incorporates a variational bottleneck that regularizes the latent space, making it a suitable candidate for evaluating the effectiveness of the Tabacof attack.
2. **Target Selection**:
    - For each input image, a target image was selected. The target image could be chosen randomly from a different class or manually specified. The goal was to generate perturbations that would make the VIB model interpret the input image as the target image.
3. **Latent Space Manipulation**:

- The attack optimizes adversarial perturbations by minimizing a combined loss function:

$$\text{Loss} = \text{KL}(\mu(x) \parallel \mu(t)) + \lambda \cdot \; L_2\_\text{norm}(x - x_{adv}) \tag{4.1}$$

  Where $\mu(x)$ and $\mu(t)$ are the mean vectors of the latent representations for the original and target images, respectively, and λ is a regularization parameter controlling the perturbation magnitude. The adversarial noise was initialized with small random values and iteratively updated to minimize the loss function.

4.2 Data Collection and Preparation

Our data collection and preparation process is crucial for ensuring the integrity and effectiveness of our experiments. Most of our selected datasets are provided directly via the torchvision.datasets library [33]. For the training set, images are augmented using several transformations to enhance data variation and improve model generalization. These transformations include random cropping with padding, random horizontal flipping, and random rotation.

Each image is then converted to a tensor and normalized using the dataset's mean and standard deviation statistics. The test set undergoes resizing, tensor conversion, and normalization to ensure consistency with the training set. The data is then loaded into pytorch Dataloaders, which facilitate batch processing and shuffling. To train our selected ResNet models, we initialize them using pre-defined architectures from torchvision.models. The fully connected layer of each model is adjusted to match the number of classes in the dataset, and the models are moved to the GPU using Cuda for accelerated training. Once the models are trained, they are saved to disk for future use. This step ensures we can reload the models without retraining, thus saving computational resources.

Dataset	**Gaussian Noise**	**Defocus Blur**	**Motion Blur**	**Low Contrast**
MNIST	SD: 0.5	Radius: 1.5	5x5 Kernel	Factor: 0.1
CIFAR-10	SD: 0.025	Radius: 2.0	5x5 Kernel	Factor: 0.25
SVHN	SD: 0.025	Radius: 2.0	5x5 Kernel	Factor: 0.25
ImageNet64	SD: 0.1	Radius: 1.25	5x5 Kernel	Factor: 0.75
Felidae	SD: 0.1	Radius: 2.0	5x5 Kernel	Factor: 0.5

Table 4.1: Distortion parameters applied to different datasets.

When generating the corrupted datasets mentioned in the section on Image Perturbations, we took a trial-and-error approach to selecting the extent of the damage. The goal was to generate noticeably altered images that would remain recognizable to a human observer. This was challenging methodically, as the severity of the corruption is subjective. Our results presented in the following chapter nonetheless reveal exciting findings. Table 4.1 outlines our selected parameters for each dataset and type of corruption.

Figures 4.5 to 4.9 showcase the generated corruptions on a sample image from each dataset, depicting from left to right: Base image, Gaussian noise, defocus blur, motion blur, and low contrast.

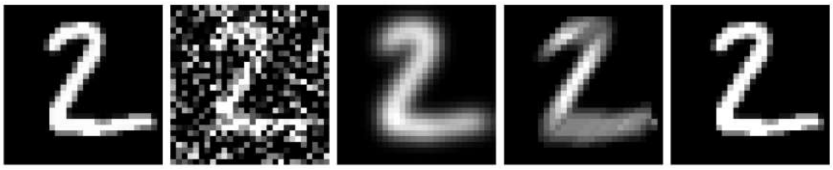


Figure 4.5: Common image corruptions applied to the MNIST dataset.

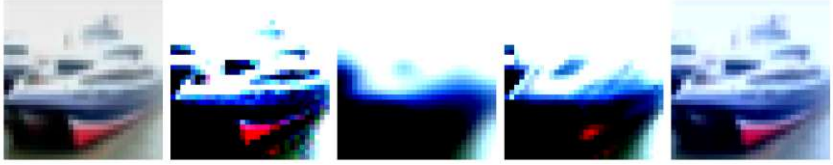


Figure 4.6: Common image corruptions applied to the CIFAR-10 dataset.

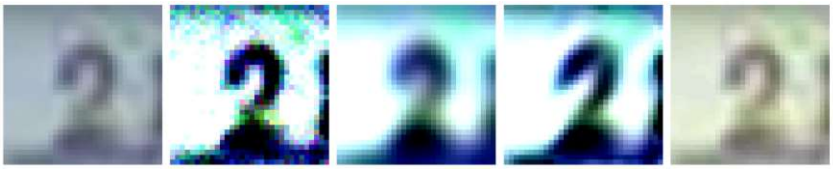


Figure 4.7: Common image corruptions applied to the SVHN dataset.

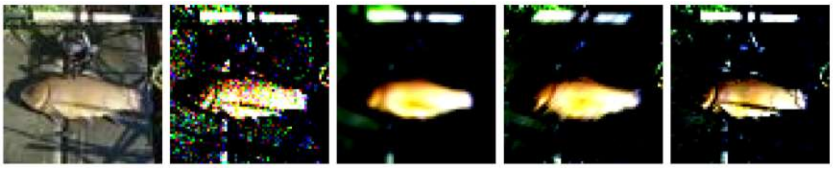


Figure 4.8: Common image corruptions applied to the ImageNet64 dataset.

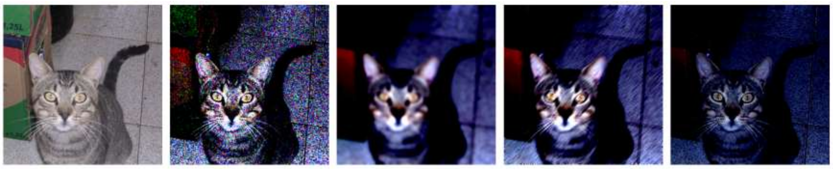


Figure 4.9: Common image corruptions applied to the Felidae dataset.

We utilize the torchattacks [20] library to generate adversarial examples. The adversarial data is generated by applying specific attacks (FGSM, EAD, C&W, JSMA/JSMAOnePixel) to the trained models. The perturbed data is then saved to disk as a torch tensor array of image tensors and labels for subsequent evaluation. Perturbed data is stored in specified directories, allowing us to reload and evaluate the models' performance under adversarial conditions without regenerating the perturbed images each time. For the most part, we have reached acceptable perturbation results using the attack parameters set as default by the individual torchattacks. The only dataset that proved challenging to perturb using these initial values was MNIST, which was classified correctly at the same rate as the base dataset under the FGSM and C&W attacks. After some experimentation, we have landed on the following adjusted parameters:

- FGSM: We have changed the maximum perturbation parameter *eps* to 32/255 from the default 8/255 for every version of ResNet

- C&W: We have changed the box-constraint parameter c to 5 from the default 1, as well as the learning rate of the Adam optimizer to 0.05 from the initial 0.01 for every version of ResNet.

Our implementation of the JSMAOnePixel has been modified slightly to include an iteration limit to prevent the attack from running excessively long, particularly when modifying many pixels is unnecessary, ensuring a balance between attack effectiveness and computational efficiency. In addition to the hard limit on iterations, we employ an early exit strategy, which stops the perturbation process as soon as the current iteration is misclassified. Usually, the perturbation process would run until the image gets misclassified to be a specific target image.

We only generate adversarial data using the base ResNet models, not injected ones. Our research aims to determine if added bottlenecks at different classification stages improve the model's ability to deal with perturbed data that base models would otherwise struggle with. Therefore, we do not want to train the attacks against these altered models. Instead, we want to simulate a scenario closer to the real world, where attacks created to fool base models would be potentially thwarted thanks to the addition of an artificial information bottleneck.

The trained models are loaded for evaluation, and their performance on clean and perturbed datasets is assessed. The specific metrics used for the assessment are outlined in the following section. This systematic approach to data collection, preparation, model training, and evaluation ensures that our experiments are thorough, reproducible, and capable of providing meaningful insights into the robustness of DNNs against adversarial attacks.

4.3 Evaluation Metrics

Our models are evaluated using a comprehensive set of metrics that provide a thorough understanding of their performance and robustness. The key metrics used in our analysis are top-1 accuracy, top-5 accuracy, bits per pixel and the L_0, L_2, and L_∞ norms. Each of these metrics serves a specific purpose and provides insights into model behavior and resilience to adversarial attacks.

Top-1 accuracy is a fundamental metric that measures the proportion of correctly classified instances out of the total cases tested. Specifically, top-1 accuracy represents the percentage of test images for which the model's highest confidence prediction matches the true label. This metric is crucial because it directly indicates the model's effectiveness in making correct predictions on clean, unperturbed data, serving as a baseline for overall performance. These statistics can then be compared with the model's performance on perturbed data to gauge the performance loss quickly.

Top-5 accuracy, on the other hand, extends this evaluation by considering the top five predictions made by the model for each test instance. If the true label is among the top five predictions, the instance is considered correctly classified for this metric. Top-5 accuracy is instrumental in applications where multiple plausible answers may exist or where it is acceptable to consider a set of potential outcomes. In our case, four out of five selected datasets only have ten classes or less, meaning the top-5 accuracy is expected to be fairly high. It is an adequate way to evaluate the correctness of our training process when we compare the two accuracies with available statistics.

In addition to accuracy metrics, we employ the L_0, L_2, and L_∞ norms to quantify the perturbations introduced by adversarial attacks. Each of these norms provides a different perspective on the nature and impact of the perturbations:

- The L_0 norm counts the number of pixels the adversarial perturbation has altered. It provides insight into the sparsity of the perturbation, indicating how many individual pixel changes are needed to deceive the model. The L_0 norm is particularly useful for understanding the minimum number of changes required to alter the model's prediction, highlighting the attack's efficiency in terms of the number of modifications rather than their magnitude.

- The L_2 norm, also known as the Euclidean norm, measures the average magnitude of perturbations applied to each pixel of an image. It is computed as the square root of the squared differences between the original and perturbed images. The L_2 norm provides a sense of the overall distortion introduced by an attack, capturing the extent to which the entire image has been altered. This metric is important because it helps us understand how much change is needed to fool the model, indicating the robustness of the model against distributed perturbations.

- The L_∞ norm, or max norm, measures the maximum change applied to any single pixel in the image. It is computed as the maximum absolute difference between the pixels of the original and perturbed images. The L_∞ norm is particularly useful for understanding the impact of localized, high-intensity perturbations. This metric is critical when minor but highly concentrated changes can significantly affect the model's predictions. We can assess the model's vulnerability to extreme yet localized alterations by evaluating the L_∞ norm.

Using all three norms allows for a comprehensive analysis of adversarial robustness. The L_0 norm reveals how sparse or dense the perturbation is, the L_2 norm provides a holistic view of the overall perturbation, and the L_∞ norm highlights the worst-case scenario of pixel-level changes. These metrics offer a detailed picture of the model's resilience to adversarial attacks, from sparse, minimal alterations to widespread distortions and sharp, focused changes.

Combining top-1 and top-5 accuracy with L_0, L_2, and L_∞ norms ensures that we capture the model's general performance and its specific vulnerabilities to adversarial perturbations. Top-1 and top-5 accuracy metrics reveal how well the model performs under normal and relaxed conditions, providing insights into its overall classification capabilities. Meanwhile, the L_0, L_2, and L_∞ norms detail the model's robustness by quantifying the extent and nature of adversarial perturbations required to deceive it.

Moreover, we use the Bits Per Pixel (BPP) metric to evaluate models with bottleneck layers. BPP measures the amount of information retained or discarded through the compression process in these models. By analyzing BPP, we aim to understand the trade-offs between the compression level and model performance, particularly under adversarial conditions. The BPP metric quantifies how much information is preserved in the bottleneck layers, which is crucial for maintaining accuracy while minimizing the data footprint. By comparing BPP values across different models, we can demonstrate how varying levels of compression impact the model's robustness to adversarial attacks and its ability to generalize from the data. This metric illustrates the effectiveness of our bottleneck-injected models in balancing compression with performance.

In addition to these quantitative metrics, we generate ten random pairs of standard and perturbed images for every attack. This qualitative evaluation allows us to visually inspect the differences introduced by adversarial perturbations. While it is effectively impossible to empirically evaluate images directly due to the subjective nature of visual assessments, including these comparison images provides a tangible representation of the perturbations. These visual comparisons can highlight the subtle yet impactful changes that adversarial attacks impose on images. In the following chapter, we include some of these comparison images to demonstrate the nature and extent of the perturbations intuitively. The full statistics collection pipeline can be observed in Figure 4.10.

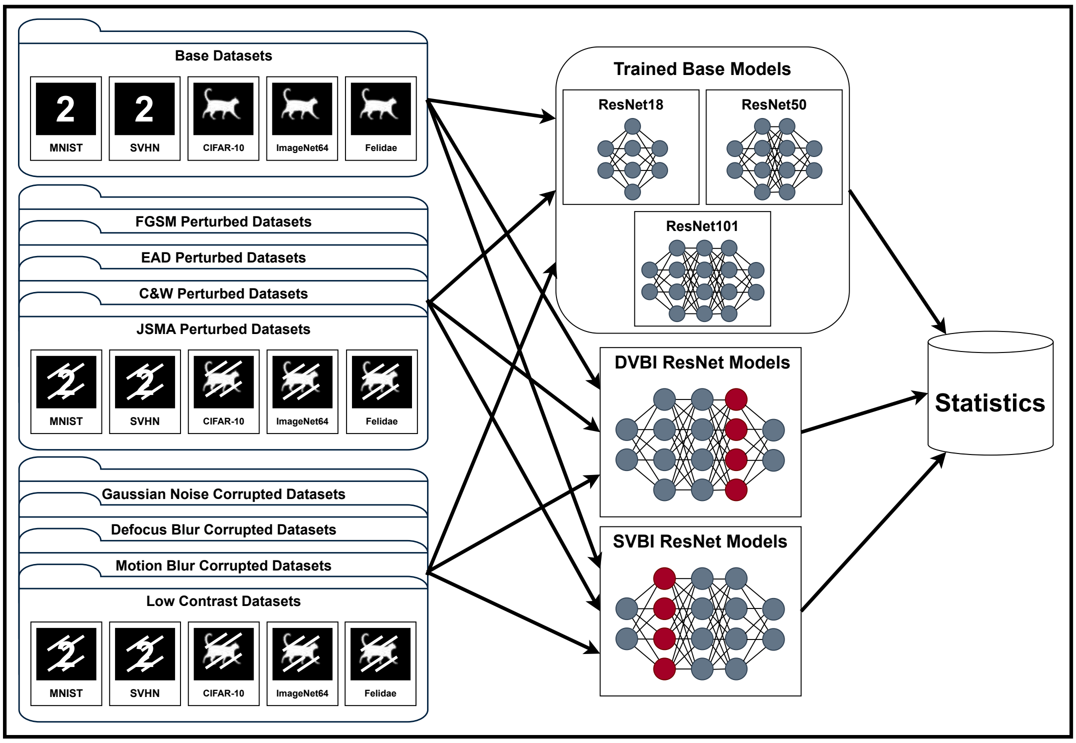


Figure 4.10: A visualization of our statistics collection process.

CHAPTER 5

Experiment Results

This chapter presents the experiment results. We analyze the effects of different image corruptions on classification performance and present statistics collected on model performance with perturbed datasets altered by specific attacks. Additionally, we examine the L-norms (L_0, L_2, and L_∞) to quantify the perturbations applied to the images. We present our results here without relating them to each other. An in-depth analysis and interpretation of the differences and effects of specific models and attacks is found in Chapter 6.

5.1 Base Accuracies of Simple Models

Through extensive experimentation, first with the ResNet-18 model for time-saving reasons, we have arrived at the following training parameters that have allowed us to train our models to mostly match the reported average benchmarks for comparable setups collected by paperswithcode.com [31]. Typical values used with all model/dataset combinations are omitted from the following list. These values include the momentum of the Stochastic Gradient Descent optimizer (when used), which was set to 0.9, and the early stopping patience of 5 epochs as well as the gradient clip value of 1.0 for all datasets except CIFAR-10 and Felidae, which was changed to 10 and 0.5 respectively. As a loss function, cross entropy loss was used during all our experiments, and a scheduler was used to reduce the learning rate on a plateau in the "min" mode, with a factor set to 0.1 and patience set to 5. The Felidae Dataset uses the OneCycleLR scheduler with a maximum learning rate of 0.01.

Dataset	Model	Optimizer	Scheduler	LR	Batch Size	Epochs	Weight Decay
MNIST	R18	SGD	ReduceLROnPlateau	0.01	128	200	0.0001
	R50	SGD	ReduceLROnPlateau	0.01	64	200	0.0001
	R101	SGD	ReduceLROnPlateau	0.01	32	200	0.0001
CIFAR-10	R18	SGD	ReduceLROnPlateau	0.0001	32	400	0.0001
	R50	SGD	ReduceLROnPlateau	0.0001	32	600	0.0001
	R101	SGD	ReduceLROnPlateau	0.0001	32	800	0.0001
SVHN	R18	SGD	ReduceLROnPlateau	0.01	128	200	0.0001
	R50	SGD	ReduceLROnPlateau	0.01	64	200	0.0001
	R101	SGD	ReduceLROnPlateau	0.01	32	200	0.0001
ImageNet64	R18	Adam	ReduceLROnPlateau	0.001	256	200	0.00001
	R50	Adam	ReduceLROnPlateau	0.001	256	200	0.00001
	R101	Adam	ReduceLROnPlateau	0.001	256	200	0.00001
Felidae	R18	AdamW	OneCycleLR	0.01	128	200	0.0001
	R50	AdamW	OneCycleLR	0.01	64	200	0.0001
	R101	AdamW	OneCycleLR	0.01	32	200	0.0001

Table 5.1: Hierarchical summary of training configurations across datasets.

The parameters used in Table 5.1 allowed us to train our baseline models to reach the accuracies reported in Table 5.2. The results mostly align with reported accuracies for these models and datasets, with some, such as CIFAR-10, being slightly lower by about 5-10 percentage points and ImageNet64 differing by about ten percentage points, especially for the deeper models. The Felidae dataset also shows relatively low accuracy, primarily due to the limited volume of training and test data available.

These slight deviations are not problematic for this work, as our goal was to observe the performance gains or losses from introducing an information bottleneck. The accuracies are sufficient to observe distinct differences in model performance with specific bottleneck injections. Most datasets were trained within two to eight hours, with the larger ImageNet64 dataset requiring over two days. This extended training time also contributed to our decision to proceed with slightly lower accuracies.

Dataset	ResNet18	ResNet50	ResNet101
MNIST	98.93%	99.12%	98.75%
CIFAR-10	88.29%	89.21%	89.05%
SVHN	94.04%	94.88%	94.37%
ImageNet64	38.05%	42.18%	42.17%
Felidae	75.43%	76.57%	79.14%

Table 5.2: Baseline accuracies of simple models on various datasets.

5.2 Effects of Image Corruptions on Simple Model Classification Accuracy

For MNIST, introducing Gaussian noise resulted in a significant drop in classification accuracy across all ResNet models, with accuracies plummeting to around 10-12% . For

Dataset	Corruption	ResNet18	ResNet50	ResNet101
MNIST	Gaussian Noise	86.79%	87.12%	**89.01%**
	Defocus Blur	18.81%	**27.92%**	22.89%
	Motion Blur	10.62%	9.83%	**10.71%**
	Low Contrast	**77.43%**	77.41%	76.72%
CIFAR-10	Gaussian Noise	0.91%	**1.15%**	0.91%
	Defocus Blur	59.76%	62.59%	**63.82%**
	Motion Blur	37.04%	**40.59%**	38.32%
	Low Contrast	**47.24%**	43.27%	41.27%
SVHN	Gaussian Noise	**0.36%**	0.19%	0.30%
	Defocus Blur	10.24%	9.34%	**10.44%**
	Motion Blur	**6.75%**	5.56%	5.73%
	Low Contrast	**11.1%**	7.07%	7.07%
ImageNet64	Gaussian Noise	31.85%	**33.52%**	33.31%
	Defocus Blur	34.94%	39.83%	**40.32%**
	Motion Blur	33.88%	37.31%	**38.21%**
	Low Contrast	**2.99%**	2.39%	2.12%
Felidae	Gaussian Noise	1.00%	19.14%	**28.00%**
	Defocus Blur	19.14%	27.14%	**30.28%**
	Motion Blur	17.72%	19.14%	**22.85%**
	Low Contrast	7.72%	11.71%	**14.00%**

Table 5.3: Decrease of simple model accuracies under different image corruptions. Bold values in each row indicate the value with the biggest drop compared to the baseline.

instance, ResNet-18's accuracy dropped sharply by around 86%. This underscores the models' high sensitivity to random noise, particularly in a simple dataset like MNIST. Notably, deeper models like ResNet-101 experienced a slightly more noticable drop than ResNet-18. Defocus blur also led to observable decreases in accuracy, with deeper models like ResNet-101 showing slightly worse resilience compared to ResNet-18. However, the impact was less severe than Gaussian noise, suggesting that spatial distortions are less detrimental than random noise. Motion blur caused moderate accuracy reductions, with ResNet-18's accuracy decreasing by 10%, while ResNet-50 and ResNet-101 maintained similar performance, indicating that temporal coherence is less disruptive, regardless of model depth. Low contrast had a critical impact on performance, with all ResNet models' accuracies dropping dramatically by around 76%, highlighting the models' dependence on sufficient contrast for feature extraction and showing little variation across model depths.

For CIFAR-10, the impact of Gaussian noise was milder compared to MNIST, with only slight drops in accuracy. For example, ResNet-18's accuracy fell by less than 1%, with ResNet-50 and ResNet-101 showing even smaller reductions. This reflects CIFAR-10's greater robustness against random noise due to its complexity. However, defocus blur led to more substantial decreases in accuracy, with all models handling this distortion similarly. Motion blur also caused significant performance drops, with ResNet-18's accuracy reducing by around 37%. At the same time, deeper models exhibited a slightly worse result, emphasizing the challenges posed by temporal distortions, even for deeper models. Low contrast had a pronounced effect, with ResNet models showing considerable accuracy reductions by more than 50%. However, deeper models like ResNet-101 show visibly better performance, indicating that contrast remains a critical factor even in more complex datasets, though depth provides some mitigation.

For SVHN, Gaussian noise resulted in minimal accuracy reductions, showing that the models retained considerable robustness to random noise, as seen in ResNet-18's minor drop by less than 1%, with deeper models like ResNet-50 and ResNet-101 exhibiting even less reduction, highlighting the small but positive impact of depth in handling noise. Defocus blur caused slightly more considerable decreases, with ResNet-18's accuracy dropping by about 10%. Deeper models performed similarly, highlighting the models' sensitivity to spatial distortions and no apparent advantage of depth. Motion blur and low contrast were also observed to be similar, with minor accuracy drops, with slight improvements in the deeper models compared to ResNet-18.

For ImageNet64, Gaussian noise caused dramatic accuracy drops, with accuracies falling to single digits, illustrating the models' extreme vulnerability to random noise in a highly complex dataset. Interestingly, ResNet-101 exhibited slightly better performance under noise, indicating that model depth can offer some limited resistance. Defocus blur had an even more severe impact, reducing ResNet-18's accuracy by almost 35%, with deeper models like ResNet-101 dropping by more than 40%, suggesting that greater depth does not necessarily confer robustness to spatial distortions in complex datasets. Motion blur and low contrast also severely affected performance, leading to an accuracy of around 4-5% across all models, showing minimal benefit from increased depth. Low contrast showed a slightly better impact, with accuracies between 35-40%, where ResNet-101 performed marginally better, highlighting that depth may offer limited benefits in maintaining performance under such conditions.

For the Felidae dataset, Gaussian noise led to moderate accuracy reductions especially in deeper models, with ResNet-101's accuracy decreasing by 28%. At the same time, the shallower ResNet-18 showed marginally better resilience, reflecting a noticeable but less severe vulnerability compared to other datasets, showing no real benefits from model depth. Defocus blur and motion blur caused more notable drops in accuracy across all model depths, however both corruptions showed a lesser impact on the shallow model. low contrast had a limited effect on all models, with only marginal differences between the models of varying depth, though the decrease from base accuracy is clear.

Our findings, presented in table 5.3, highlight the varying levels of performance degradation we measured across different image corruptions and dataset complexities, with more pronounced accuracy drops in certain conditions like Gaussian noise or low contrast.

5.3 Effects of Adversarial Attacks on Base Classification Accuracy

Dataset	Attack	ResNet18	ResNet50	ResNet101
MNIST	FGSM	**54.75%**	35.16%	44.63%
	EAD	98.01%	98.27%	**98.64%**
	C&W	**94.58%**	88.36%	93.92%
	JSMA	**94.78%**	83.63%	84.57%
CIFAR-10	FGSM	**74.56%**	68.85%	69.31%
	EAD	85.52%	85.14%	**88.65%**
	C&W	87.01%	**88.01%**	87.75%
	JSMA	**87.62%**	85.10%	83.58%
SVHN	FGSM	**69.95%**	58.06%	68.31%
	EAD	90.54%	92.09%	**94.32%**
	C&W	92.37%	88.78%	**93.23%**
	JSMA	93.58%	90.17%	**94.08%**
ImageNet64	FGSM	37.76%	41.70%	**41.71%**
	EAD	35.43%	37.73%	**41.66%**
	C&W	37.85%	41.73%	**41.94%**
	JSMA	34.66%	**39.32%**	38.97%
Felidae	FGSM	**75.43%**	72.86%	74.85%
	EAD	72.57%	75.43%	**78.28%**
	C&W	**75.14%**	66.86%	73.43%
	JSMA	74.86%	75.43%	**78.57%**

Table 5.4: Decrease of base model accuracies under different adversarial attacks. Bold values in each row indicate the value with the biggest drop compared to the baseline.

Presented in table 5.4 are the relative accuracy decreases of our base models on adversarial datasets, compared to unperturbed data. The FGSM attack, which perturbs input images by adjusting each pixel toward the loss function gradient, caused significant accuracy drops across all models and datasets. For MNIST, accuracies fell noticeably, with substantial reductions across all ResNet models. For example, ResNet-18's accuracy dropped by more than 50%, while deeper models like ResNet-50 and ResNet-101 showed relatively better resilience. This indicates that increased model depth may provide some robustness against FGSM, though the perturbations still significantly degrade performance. The L_2 and L_0 norms indicated substantial overall perturbations, with ResNet-18 showing almost 60% of pixels altered, highlighting the cumulative effect of subtle perturbations that the models failed to manage effectively. For CIFAR-10 and SVHN, similar trends were observed, with accuracy drops remaining substantial despite the higher complexity of these datasets. In CIFAR-10, the L_0 norms were considerably higher than other attacks, such as ResNet-18 showing more than 99% altered pixels, indicating that many pixels needed to be perturbed to achieve a significant drop in accuracy. In ImageNet64, FGSM also led to notable accuracy reductions, with ResNet-18 showing a drop by more than 35%, while deeper models such as ResNet-50 and ResNet-101 fared only marginally better. The L_0 norm showed more than 98% altered pixels in ResNet-18 reflecting the challenge of maintaining robustness against such attacks in high-resolution images. The L_∞ norms, although lower, still demonstrated impactful pixel-wise changes. The Felidae dataset also experienced significant accuracy reductions under FGSM, with extraordinarily high L_0 norms, such as ResNet-18 showing more than 99% of pixels altered, indicating that nearly the entire image was altered. ResNet-50 and ResNet-101, with accuracy drops of more than 70%, managed to handle the perturbations slightly better than ResNet-18, again suggesting that deeper models can offer improved, though still limited, resilience. The high L_0 statistics are mirrored in the high L_∞ norms indicating that in addition to the amount of altered pixels, significant pixel-wise changes were necessary to cause notable degradation in model performance.

A sample set of the FGSM perturbed images can be found in the appendix section A.1.

EAD resulted in even more significant accuracy reductions than FGSM across almost all datasets. The L_0 norms were higher, with, for instance, ResNet-18 showing almost 93% altered pixels in MNIST, reflecting the increased complexity of the perturbations. CIFAR-10 also saw substantial drops, with ResNet-18's accuracy further reduced by more than 85%. Interestingly, ResNet-50 exhibited slightly better resilience, while ResNet-101 struggled more significantly, dropping almost to zero. This counterintuitive result may arise from the deeper model's greater complexity, sometimes making it more sensitive to subtle perturbations or from optimization and gradient stability challenges in very deep networks. SVHN followed a similar pattern, with ResNet-18's accuracy dropping by around 90% under EAD, while ResNet-50 and ResNet-101 exhibited even lower accuracies, indicating that the increased model depth did not confer additional robustness in this case. The L_0 norms were also high suggesting that the attack effectively exploited the vulnerabilities of deeper models in this dataset. In ImageNet64, the EAD attack also led to significant accuracy reductions. ResNet-50 and ResNet-101 demonstrated a similar trend to the previous dataset. However, the overall impact remained severe, reflecting the severity of perturbations required to compromise the models in this high-resolution dataset. The L_∞ norms, though comparably lower, indicated that maximum pixel-wise changes still considerably impacted accuracy, regardless of model depth. For the Felidae dataset, EAD caused notable accuracy drops, with moderate L_2 and L_0 norms underscoring the extensive alterations necessary to affect the model's performance in this complex subset, reducing the accuracy of ResNet-101 to less than 1%. ResNet-50 and ResNet-18 notably both performing slightly better.

A sample set of the EAD perturbed images can be found in the appendix section A.2.

The C&W attack, known for optimizing perturbations to maximize misclassification while minimizing detection, resulted in the most severe accuracy reductions across all datasets. For example, in MNIST, ResNet-18's accuracy plummeted by around 94%, with an L_0 norm showing around 18% pixels altered, demonstrating the attack's effectiveness even with minimal pixel changes. Deeper models like ResNet-101 and especially ResNet-50 exhibited slightly better performance, indicating that while C&W is highly effective, model depth can offer some resistance. CIFAR-10 and ImageNet64 exhibited similar trends, with substantial accuracy drops across the board, reflecting the sophisticated nature of this attack and the marginal benefits of increased model depth. SVHN also saw severe accuracy reductions, while the L_0 norms for all datasets except MNIST turned out higher than expected. They were, however, balanced out by lower L_∞ values. In the Felidae dataset, C&W attacks showed a similar trend, with ResNet-18 showing an L_0 norm of above 99% of pixels altered, highlighting the extent of perturbations needed to mislead the model in this subset, with accuracy dropping to nearly zero for ResNet-18. ResNet-50 and ResNet-101 fared noticeably better in this case. A sample set of the C&W perturbed images can be found in the appendix section A.3.

JSMA perturbs the input by modifying only the most influential pixels, leading to significant accuracy reductions across all datasets while achieving higher overall L_2 and L_∞ norms. In MNIST, ResNet-18 showed an L_0 norm of less than 7% pixels altered, indicating that JSMA effectively causes misclassification with minimal pixels changed. ResNet-50 and ResNet-101 showed better resilience with accuracies dropping by around 84%, compared to 94% for ResNet-18, highlighting that deeper models are better equipped to handle such targeted perturbations. This trend was consistent across CIFAR-10 and Felidae, where minimal but highly effective perturbations led to notable accuracy drops, reducing ResNet-18's accuracy to nearly zero in both CIFAR-10 and Felidae, with ResNet-50 showing better performance, reflecting the models' vulnerabilities to targeted pixel-wise changes and the benefits of increased depth. SVHN and ImageNet64 also experienced notable reductions under JSMA, with ResNet-50's accuracy being slightly higher than both the shallower and deeper counterparts in both datasets.

A sample set of the JSMA perturbed images can be found in the appendix section A.4. Overall, the analysis reveals that all tested adversarial attacks, particularly C&W and EAD, significantly degrade the performance of ResNet models across various datasets, with the most complex attacks (C&W) showing the highest effectiveness even with minimal perturbations. However, deeper ResNet models (50 and 101) consistently show slightly better resilience across different attacks and datasets, indicating that while no model is immune, increased depth provides incremental improvements in robustness. These results were expected, and the evident degradation of classification accuracy provides insightful comparisons to the bottleneck-injected models analyzed in the following sections.

A summary of the norms we measured is presented in Tables 5.5 to 5.7.

Dataset	Attack	L0 Norm	L2 Norm	Linf Norm
MNIST	FGSM	461.5096	2.5821	0.1255
	EAD	728.1548	1.2548	0.4337
	C&W	141.6285	2.3910	0.5371
	JSMA	54.0691	4.6363	0.9996
CIFAR-10	FGSM	3056.0245	1.7233	0.0314
	EAD	2755.8611	0.3302	0.0954
	C&W	2827.0205	2.3598	0.5379
	JSMA	690.7737	5.2796	0.9996
SVHN	FGSM	3064.6276	1.7323	0.0314
	EAD	2898.3939	0.4535	0.1240
	C&W	2928.2497	1.7324	0.0314
	JSMA	597.4976	2.5563	0.7163
ImageNet64	FGSM	12124.4091	3.4254	0.0314
	EAD	6702.5966	0.0907	0.0267
	C&W	8376.4446	2.3018	0.5484
	JSMA	2986.9732	2.3569	0.3423
Felidae	FGSM	149255.3029	104.4099	0.9268
	EAD	149561.6486	103.8224	0.9146
	C&W	149747.7514	103.8156	0.9140
	JSMA	148577.6086	104.0326	0.9445

Table 5.5: L0, L2, and Linf Norms for ResNet18 Under Different Adversarial Attacks.

Dataset	Attack	L0 Norm	L2 Norm	Linf Norm
MNIST	FGSM	433.0476	2.4852	0.1255
	EAD	728.1548	1.3020	0.4630
	C&W	136.6747	2.3598	0.5379
	JSMA	62.6740	5.2796	0.9996
CIFAR-10	FGSM	3056.0256	1.7234	0.0314
	EAD	2760.9355	0.3398	0.0953
	C&W	2842.4857	2.3598	0.5379
	JSMA	709.2510	3.9341	0.8541
SVHN	FGSM	3064.6545	1.7324	0.0314
	EAD	2905.1410	0.4982	0.1309
	C&W	2843.8885	0.4529	0.0414
	JSMA	612.7123	3.1362	0.7219
ImageNet64	FGSM	12124.3901	3.4254	0.0314
	EAD	7337.2006	0.0936	0.0294
	C&W	8600.3832	0.1741	0.0054
	JSMA	2990.6234	2.4747	0.3749
Felidae	FGSM	149231.4771	104.3837	0.9268
	EAD	149664.5314	103.8202	0.9146
	C&W	149660.88	103.8184	0.9141
	JSMA	148577.8114	104.0299	0.9441

Table 5.6: L0, L2, and Linf Norms for ResNet50 Under Different Adversarial Attacks.

Dataset	Attack	L0 Norm	L2 Norm	Linf Norm
MNIST	FGSM	430.9440	2.4805	0.1255
	EAD	743.899	1.1664	0.4339
	C&W	141.0340	2.3018	0.5484
	JSMA	60.8919	5.1902	0.9994
CIFAR-10	FGSM	3055.9931	1.7234	0.0314
	EAD	2754.4508	0.3453	0.0943
	C&W	2838.7128	2.3598	0.5379
	JSMA	710.4651	4.0352	0.8588
SVHN	FGSM	3064.5567	1.7323	0.0314
	EAD	2858.4523	0.4675	0.1292
	C&W	2943.2363	0.4665	0.0407
	JSMA	600.5013	2.7478	0.7177
ImageNet64	FGSM	12124.2710	3.4254	0.0314
	EAD	7143.7339	0.0912	0.0290
	C&W	8620.3321	0.1742	0.0052
	JSMA	2991.3329	2.5026	0.3772
Felidae	FGSM	149239.8229	104.3879	0.9276
	EAD	149581.6229	103.8192	0.9146
	C&W	149730.6114	103.8181	0.9141
	JSMA	148577.9286	103.9931	0.9447

Table 5.7: L0, L2, and Linf Norms for ResNet101 Under Different Adversarial Attacks.

5.4 Base Accuracies of DVBI Models

We trained new DVBI models on various datasets using the same pipeline and training parameters used for the base models, ensuring an unbiased comparison. Interestingly, the DVBI models almost consistently reach a lower baseline accuracy than the base models. Except for the ImageNet64 dataset tested on ResNet-50, and SVHN with ResNet-101, the accuracy differences are generally lower by almost five percentage points across the board. The slight decrease in accuracy observed in DVBI ResNet models compared to the base ResNet models could be attributed to the compression-accuracy trade-off introduced by the deep bottleneck mechanism [39]. While the bottleneck layer aims to reduce the size of feature representations, this compression may lead to a loss of discriminative power, limiting the model's ability to capture the full richness of features necessary for optimal classification. Additionally, the bottleneck can act as an over-regularization, constraining the flow of information and reducing the model's capacity to capture fine-grained details. This trade-off between compressibility and accuracy, along with a narrower feature diversity, is likely responsible for the slight underperformance observed in the DVBI ResNet models.

The baseline results indicate that the DVBI models achieve high accuracy on MNIST and SVHN datasets across all ResNet variants. For CIFAR-10, the ResNet-18 model demonstrates the highest top-1 accuracy, while the ResNet-50 model performs slightly better than ResNet-101. On the more challenging ImageNet64 dataset, the top-1 accuracies are lower, with the ResNet-50 and ResNet-101 models performing better than ResNet-18. The top-1 accuracy with the Felidae dataset varies, with ResNet-50 achieving the highest accuracy, even compared to the base model. This discrepancy is probably the result of the limited training data for this dataset.

The BPP metric played a crucial role in training the DVBI models. We systematically tested several values for the λ parameter mentioned in Section 2 and tried to achieve an acceptable balance of classification accuracy and Bits Per Pixel. We tested values in the range from 0 to 1000, using starting small and potentially increasing the number of datapoints if the accuracy kept reaching acceptable results. The following table (5.8) depicts the BPP values we reached for our trained DVBI models. The reported

Dataset	ResNet18	ResNet50	ResNet101
MNIST	0.015	0.012	0.021
CIFAR-10	0.065	0.012	0.015
SVHN	0.0075	0.008	0.01
ImageNet64	0.078	0.079	0.077
Felidae	0.0002	0.0003	0.0005

Table 5.8: BPP values for different datasets and DVBI ResNet architectures.

BPP values highlight the efficiency of different ResNet architectures in compressing features across various datasets. ResNet-18 shows the lowest BPP values for Felidae (0.0002), indicating strong compression efficiency. In contrast, the ImageNet64 and CIFAR-10 datasets require significantly more bits per pixel, reflecting their apparent higher complexity. For ImageNet64, BPP values are consistent across all models, around 0.078, indicating similar compression performance. The BPP values vary by model and dataset, showing that compression efficiency is context-dependent.

5.5 Effects of Image Corruptions on DVBI Model Classification Accuracy

The impact of various image corruptions on the DVBI models is summarized in Table 5.9. The results demonstrate that image corruptions, particularly Gaussian noise and defocus blur, significantly degrade the models' performance.

Dataset	Corruption	ResNet18	ResNet50	ResNet101
MNIST	Gaussian Noise	80.30%	**85.28%**	74.28%
	Defocus Blur	**78.17%**	29.17%	48.97%
	Motion Blur	39.30%	22.81%	**43.06%**
	Low Contrast	**85.58%**	80.39%	84.62%
CIFAR-10	Gaussian Noise	1.22%	1.32%	**1.42%**
	Defocus Blur	62.08%	63.81%	**64.44%**
	Motion Blur	39.42%	**44.29%**	38.65%
	Low Contrast	54.33%	**56.10%**	51.04%
SVHN	Gaussian Noise	0.22%	**0.29%**	0.26%
	Defocus Blur	**10.51%**	10.12%	10.35%
	Motion Blur	**6.75%**	6.14%	5.75%
	Low Contrast	8.86%	**11.81%**	11.60%
ImageNet64	Gaussian Noise	29.46%	**34.82%**	33.41%
	Defocus Blur	35.23%	38.85%	**38.92%**
	Motion Blur	32.80%	**37.45%**	36.31%
	Low Contrast	2.63%	**2.84%**	2.55%
Felidae	Gaussian Noise	10.86%	**17.72%**	17.71%
	Defocus Blur	14.29%	**18.86%**	15.71%
	Motion Blur	11.14%	**14.29%**	7.43%
	Low Contrast	9.14%	9.15%	**13.14%**

Table 5.9: Decrease of DVBI model accuracies under different image corruptions. Bold values in each row indicate the value with the biggest drop compared to the baseline.

Gaussian noise significantly reduces the accuracy of all DVBI models, highlighting their vulnerability to random noise. On the MNIST dataset, ResNet-18 drops by around 80%, ResNet-50 by around 84%, and ResNet-101 by almost 75%. ResNet-101 shows slightly better resilience than the other models, possibly due to its increased depth, but overall performance remains poor across all of them. Defocus blur also shows varying impacts across model depths. ResNet-50 achieves the highest accuracy on MNIST with only a 30% reduction, while ResNet-18 drops by almost 80%, suggesting that mid-depth models like ResNet-50 may balance complexity and overfitting more effectively when handling spatial distortions. Motion blur also affects the models differently, with ResNet-50 showing the highest resilience at a 23% reduction, followed by ResNet-18 with a decrease of around 40%. ResNet-50's architecture appears better suited to mitigating the disruptive effects of motion blur. Low contrast caused the largest accuracy decrease across all three models, underlining the models' reliance on contrast when working with the MNIST dataset.

On CIFAR-10, Gaussian noise has less impact, with ResNet-18 dropping by around 1% in accuracy, slightly higher than ResNet-50 and ResNet-101. Defocus and motion blur, however, degrade performance significantly, particularly for ResNet-101, which drops by almost 65% and 40% respectively. ResNet-18 performs slightly better under both corruptions, suggesting that deeper models may be more vulnerable to spatial and temporal distortions. Low contrast affects all models, but ResNet-18 and ResNet-101 perform similarly, slightly outperforming ResNet-50. This suggests that contrast sensitivity does not notably improve with deeper architectures.

On SVHN, Gaussian noise seems to have no notable effect across model depths, with the accuracy only decreasing by around 1% with all models. Defocus blur, motion blur and low contrast degrade performance more noticeably, with the deeper models generally outperforming ResNet-18, suggesting that deeper architectures may deal better with noise and blur. Low contrast displays the opposite trend, where ResNet-18 decreased in accuracy by around 8% and ResNet-50 dropping by around 12%. A small but notable difference.

The ImageNet64 dataset poses a significant challenge under corruption. Gaussian noise causes severe accuracy drops, with ResNet-18 decreasing by around 30%, and defocus blur further reduces accuracies by as much as 35% for ResNet-18. Even under low contrast, ResNet-101 slightly outperforms others with a 2% reduction, but the overall low accuracies indicate that DVBI models struggle considerably with corrupted high-resolution images regardless of depth. This suggests that while deeper models might offer slight improvements in handling certain corruptions, the complexity of ImageNet64 makes these improvements marginal.

For the Felidae dataset, Gaussian noise and other corruptions reduce performance across all models, but the deeper models consistently demonstrate somewhat higher resilience. This superior performance suggests that in more complex, real-world datasets like Felidae, the depth of ResNet-50 and ResNet-101 provides a more substantial benefit in coping with challenging corruptions, likely due to its enhanced ability to capture intricate features. However, overall performance is still compromised compared to unperturbed conditions, indicating that while depth helps, it is not the solution for corruption robustness.

When comparing simple models with DVBI models across various types of image corruptions, distinct patterns emerge, showcasing the potential and limitations of DVBI. Under Gaussian noise, DVBI models generally perform better or comparably on most datasets, with notable gains on MNIST and the Felidae dataset, particularly in deeper models like ResNet-101. However, some drops are observed, such as with ResNet-50. Defocus blur reveals uneven impacts; while DVBI models struggle significantly on MNIST with a steep accuracy drop in ResNet-18, they show improvements on the Felidae dataset, especially for ResNet-50. Motion blur affects both model types similarly but often results in DVBI underperforming slightly, except on the Felidae dataset, where ResNet-101 demonstrates clear robustness gains. Low contrast corruption proves more challenging for DVBI models, notably on simpler datasets like MNIST and CIFAR-10, where deeper models like ResNet-101 show marked declines, though performance remains relatively stable on SVHN and marginally better for ResNet-50 on the Felidae dataset. Overall, while DVBI models tend to underperform against various corruptions, particularly in simpler datasets, they occasionally excel in specific scenarios, especially with deeper models on complex datasets, indicating potential for refinement to enhance robustness consistently across all conditions.

5.6 Effects of Adversarial Attacks on DVBI Classification Accuracy

The effects of various adversarial attacks on the DVBI models are detailed in Table 5.10. The table illustrates the models' performance under FGSM, EAD, C&W, and JSMA attacks.

Dataset	Attack	ResNet18	ResNet50	ResNet101
MNIST	FGSM	13.34%	17.22%	**19.29%**
	EAD	7.13%	**12.79%**	8.71%
	C&W	9.73%	**20.20%**	14.09%
	JSMA	34.42%	**38.54%**	33.67%
CIFAR-10	FGSM	**39.52%**	37.84%	37.62%
	EAD	2.85%	**4.61%**	3.62%
	C&W	4.76%	**6.56%**	5.72%
	JSMA	17.57%	19.63%	**21.23%**
SVHN	FGSM	42.31%	42.09%	**47.94%**
	EAD	8.34%	**12.76%**	9.78%
	C&W	10.20%	**13.67%**	12.49%
	JSMA	21.73%	30.22%	**32.23%**
ImageNet64	FGSM	28.59%	**32.18%**	31.74%
	EAD	1.13%	**1.15%**	1.04%
	C&W	1.96%	**2.56%**	2.53%
	JSMA	9.00%	10.25%	**11.12%**
Felidae	FGSM	27.14%	**34.86%**	32.86%
	EAD	3.72%	**8.57%**	7.14%
	C&W	3.43%	**9.43%**	4.28%
	JSMA	4.29%	**8.57%**	5.43%

Table 5.10: Decrease of DVBI model accuracies under different adversarial attacks. Bold values in each row indicate the value with the biggest drop compared to the baseline.

FGSM generally causes a significant drop in accuracy across all datasets, but the extent of this drop varies by architecture and dataset. On MNIST, ResNet-18 shows strong resilience with an accuracy decrease of only around 13%, while deeper models like ResNet-50 and ResNet-101 experience more pronounced drops by around 17% and 20%, respectively. This suggests that while ResNet-18 maintains better robustness under FGSM, deeper models may be more vulnerable. On CIFAR-10, all models experience a substantial drop, revealing a consistent vulnerability. The impact is also fairly severe on SVHN, with ResNet-101 dropping sharply by amost 50%, whereas ImageNet64 shows slightly better relative resilience, with ResNet-101 dropping by around 30%.

EAD is particularly effective across most datasets, causing notable reductions in accuracy. On MNIST, ResNet-18 retains high accuracy, dropping only by around 7%, but deeper models see reduced performance, indicating that depth does not necessarily provide better protection. A similar pattern is observed on CIFAR-10 and SVHN, with ResNet-50 dropping by around 13% on the latter, compared to ResNet-18 which drops by 8%. Interestingly, the accuracy picks up again with ResNet-101. In ImageNet64, ResNet-50 slightly outperforms ResNet-18. On the Felidae dataset, however, ResNet-50 achieves the highest accuracy, suggesting some resilience in more complex datasets.

The C&W attack significantly degrades model performance across datasets. On MNIST, ResNet-18 maintains relatively high accuracy, while ResNet-50 and ResNet-101 drop by roughly 20% and 15%, respectively. This indicates that ResNet-18 shows competitive performance, highlighting the attack's varied impact. A similar trend is seen on CIFAR-10, where ResNet-50 and ResNet-101 struggle more, both dropping by around 7%. On Felidae, however, all models retain high accuracy, with ResNet-101 experiencing the smallest accuracy drop of only around 4%, again demonstrating that deeper models may offer some advantage in complex datasets.

JSMA has a more drastic impact on model accuracy. On MNIST, ResNet-18 shows some robustness with a relatively small drop of 33%, but ResNet-50 and ResNet-101 drop noticeably an average of 36%. On CIFAR-10, ResNet-18 holds a higher accuracy compared to ResNet-50. For SVHN, ResNet-101 drops sharply by almost 34%, reflecting greater vulnerability. In contrast, on the Felidae dataset, all models, including ResNet-101, maintain higher accuracies, with ResNet-101 only achieving a small drop of around 5%

These results suggest that DVBI models exhibit varying levels of robustness depending on the attack and dataset. Deeper models tend to struggle more against attacks like JSMA, which exploits pixel dependencies. However, deeper models show greater resilience in more complex datasets like Felidae, indicating that both model architecture and dataset complexity play important roles in determining adversarial vulnerability.

5.7 Base Accuracies of SVBI Models

The SVBI models exhibited unperturbed baseline accuracies that were generally even lower than those of the DVBI models across most datasets, usually only by 1-3%, though in some cases by almost 10%. As with DVBI, the base models' pipeline and training parameters were kept for a fair comparison. Notably, on simpler datasets like MNIST, all models achieve high accuracy, with ResNet-50 reaching the highest accuracy among the models. ResNet-18 and ResNet-101 follow closely, suggesting that the shallow variational bottleneck may effectively retain sufficient representational power for simpler classification tasks, even with increased model depth.

On slightly more complex datasets, the accuracies demonstrate more variability, especially for CIFAR-10 and SVHN. In the CIFAR-10 dataset, ResNet-18 attains the highest top-1 accuracy, outperforming both ResNet-50 and ResNet-101. This trend suggests that for CIFAR-10, the increased depth and parameterization of ResNet-50 may overfit or lead to reduced generalization with the shallow bottleneck injected. Similarly, on SVHN, the performance is highest with ResNet-50, slightly outperforming ResNet-18, while ResNet-101 exhibits a small drop. The results on ImageNet64 show a marked decrease in accuracy across all models, with marginal improvements from ResNet-50 over ResNet-18 and ResNet-101. This trend suggests that the shallow bottleneck's capacity may be constrained on datasets requiring richer feature hierarchies, with minimal performance gain from increased model depth. The Felidae dataset, however, shows the opposite effect, where the two deeper ResNet models fare the best. Table 5.11 presents the bits-per-pixel values for different ResNet models, indicating the models' average information density per pixel. As a reminder, lower BPP values suggest greater compression, while higher values indicate increased data requirements. For example, MNIST shows fairly low BPP values across all models, with ResNet-101 having the lowest (0.063), reflecting the dataset's simplicity. BPP values of the Felidae dataset seem to be about no par with the rest. In contrast, ImageNet64 demonstrates higher BPP values, particularly for ResNet-50 and ResNet-101, highlighting the greater complexity and feature richness needed to represent this dataset effectively.

Dataset	ResNet18	ResNet50	ResNet101
MNIST	0.076	0.111	0.063
CIFAR	0.831	0.384	0.488
SVHN	0.583	0.411	0.304
ImageNet64	1.285	1.860	1.858
Felidae	0.105	0.308	0.249

Table 5.11: BPP values for different datasets and SVBI ResNet architectures.

5.8 Effects of Image Corruptions on SVBI Model Classification Accuracy

The impact of various image corruptions on the SVBI models is summarized in Table 5.12. The results show that image corruptions, especially defocus blur and low contrast, have a varied impact on the performance of SVBI models, depending on model depth and the specific corruption.

Dataset	Corruption	ResNet18	ResNet50	ResNet101
MNIST	Gaussian Noise	0.09%	0.04%	**0.15%**
	Defocus Blur	26.97%	**62.75%**	46.16%
	Motion Blur	11.10%	14.89%	**18.57%**
	Low Contrast	70.17%	58.91%	**83.50%**
CIFAR-10	Gaussian Noise	1.28%	**2.60%**	2.32%
	Defocus Blur	**56.84%**	53.27%	55.55%
	Motion Blur	34.52%	35.08%	**39.51%**
	Low Contrast	52.33%	54.08%	**55.80%**
SVHN	Gaussian Noise	0.26%	0.36%	**0.39%**
	Defocus Blur	10.18%	9.97%	**11.46%**
	Motion Blur	**6.52%**	6.12%	5.83%
	Low Contrast	10.64%	13.05%	**14.73%**
ImageNet64	Gaussian Noise	33.11%	**35.22%**	33.09%
	Defocus Blur	33.00%	**34.99%**	34.00%
	Motion Blur	26.10%	25.56%	**26.23%**
	Low Contrast	**3.92%**	3.78%	3.76%
Felidae	Gaussian Noise	**8.86%**	6.28%	6.28%
	Defocus Blur	**17.43%**	17.14%	14.00%
	Motion Blur	12.58%	**13.14%**	**13.14%**
	Low Contrast	30.29%	34.57%	**36.86%**

Table 5.12: Decrease of SVBI model accuracies under different image corruptions. Bold values in each row indicate the value with the biggest drop compared to the baseline.

On MNIST, SVBI models are highly resilient to Gaussian noise, retaining accuracy levels above 97% across all model depths. Defocus blur, however, significantly impacts performance. ResNet-18 maintains better stability here, while deeper models see larger accuracy decreases. Motion blur is moderately disruptive, with accuracy levels dropping by around 18% in the deeper models, showing that SVBI models retain reasonable accuracy against this corruption. Low contrast proves challenging, with ResNet-101 dropping by roughly 90%, underscoring a general sensitivity to contrast reduction.

On CIFAR-10, defocus and motion blur lead to notable declines, particularly for the deeper models. ResNet-18 performs slightly better, suggesting simpler architectures may be less vulnerable to these spatial distortions. Defocus blur impacts all models similarly, compared to motion blur and low contrast, where ResNet-18 performed noticeably better than the deeper models, generally by around 10%. In the SVHN dataset, Gaussian noise has only a minor effect, with models retaining accuracy levels in the low 90% range, indicating strong resilience. Defocus and motion blur degrade performance but remain manageable, with accuracies dropping by an average of 13% across model depths. Low contrast affects SVHN models less severely, suggesting that this dataset's features may be less sensitive to contrast variations. For ImageNet64, SVBI models face substantial challenges under corruption, with Gaussian noise dropping accuracy by as much as 35%.

Defocus blur similarly reduces accuracy, while low contrast has a comparatively smaller effect, though still low for practical purposes. The Felidae dataset handles the Gaussian noise corruption reasonably well, only suffering from a roughly 10% decrease in accuracy across model depths. The remaining corruptions lower the accuracy of all models, with ResNet-101 dropping by around 37% on the low-contrast images. Depth seems to make a difference for Gaussian noise, defocus blur and motion blur, where the deepest models performed noticeably better.

The comparison of DVBI and SVBI models under various image corruptions highlights notable differences in robustness across datasets and corruption types, emphasizing the strengths and weaknesses of each approach. SVBI models demonstrate substantial resilience to Gaussian noise, maintaining accuracy above 90% across all depths on datasets like MNIST and SVHN, while DVBI models suffer significant drops, with MNIST accuracies falling by as much as 80%. On the Felidae dataset, SVBI models outperform DVBI by an average of 5%, showcasing their superior handling of random noise across most datasets. Under defocus blur, however, the advantage shifts; DVBI models, particularly shallower architectures, perform more consistently, outpacing SVBI on the Felidae dataset by about 5%, although SVBI models like ResNet-50 on MNIST occasionally stand out. Motion blur further underscores this dichotomy, with DVBI excelling on complex datasets like Felidae, where ResNet-101 achieves a 10% accuracy boost over SVBI, while SVBI proves more robust on simpler datasets like MNIST, maintaining higher accuracy across depths. Low contrast corruption similarly splits the results: SVBI retains stronger performance on simpler datasets like MNIST, with ResNet-50 achieving a 25% accuracy advantage over DVBI, whereas DVBI outperforms on high-resolution datasets like ImageNet64, where ResNet-101 surpasses its SVBI counterpart by 8%. In summary, SVBI models excel in robustness against Gaussian noise and on simpler datasets, while DVBI models occasionally outperform in handling defocus blur, motion blur, and low contrast in complex or high-resolution scenarios, suggesting their suitability depends on dataset complexity and corruption type.

5.9 Effects of Adversarial Attacks on SVBI Classification Accuracy

Unsurprisingly, the impact of adversarial perturbations on SVBI models again varies across datasets and attack types, with different levels of resilience depending on model depth and the specific adversarial method used. The specifics can be observed from table 5.13.

Dataset	Attack	ResNet18	ResNet50	ResNet101
MNIST	FGSM	5.52%	3.70%	**5.76%**
	EAD	**3.87%**	3.64%	3.84%
	C&W	**5.91%**	5.60%	5.31%
	JSMA	13.65%	**19.78%**	16.86%
CIFAR-10	FGSM	**48.75%**	24.72%	34.08%
	EAD	3.96%	4.52%	**4.88%**
	C&W	**7.77%**	5.42%	6.33%
	JSMA	**20.78%**	17.87%	17.64%
SVHN	FGSM	**51.84%**	44.48%	47.59%
	EAD	7.36%	7.23%	**8.96%**
	C&W	**12.86%**	10.94%	11.85%
	JSMA	16.08%	**20.28%**	19.39%
ImageNet64	FGSM	**32.72%**	32.22%	31.13%
	EAD	**1.96%**	1.81%	1.84%
	C&W	5.48%	5.52%	**5.53%**
	JSMA	17.07%	**19.18%**	15.39%
Felidae	FGSM	**24.58%**	22.86%	22.57%
	EAD	**7.72%**	4.28%	4.57%
	C&W	**7.43%**	4.28%	4.86%
	JSMA	**8.00%**	4.00%	5.43%

Table 5.13: Decrease of SVBI model accuracies under different adversarial attacks. Bold values in each row indicate the value with the biggest drop compared to the baseline.

On MNIST, SVBI models show high resilience to adversarial attacks, maintaining accuracy around 7% lower than baseline under FGSM, EAD, and C&W attacks. ResNet-50 performs slightly better across most attacks. However, JSMA poses a greater challenge, reducing accuracy by roughly 18%, suggesting that this method disrupts MNIST classification more effectively than others.

On CIFAR-10, SVBI models experience significant accuracy drops under FGSM, with ResNet-18 dropping by roughly 50% and ResNet-50 with a smaller drop of around 24%. For EAD and C&W attacks, however, resilience is notably higher. JSMA has a moderate impact, lowering the accuracy by almost 20% in ResNet-50. These results indicate that CIFAR-10 classifications are more vulnerable to gradient-based attacks like FGSM but show better resilience under optimization-based attacks like EAD and C&W.

The SVHN dataset shows similar trends, with a stark accuracy drop under FGSM, especially for ResNet-18 and ResNet-101, which both fall by more than 45%. ResNet-50 shows slightly better resilience. Under EAD, however, all SVBI models maintain high accuracy above 83%, reflecting strong robustness to this attack on SVHN. C&W also sees respectable accuracy, with all models performing only around 10% worse than baseline. JSMA remains somewhat effective, lowering accuracy by roughly 20%.

For ImageNet64, SVBI models demonstrate significant vulnerability to all attacks, especially FGSM, which reduces accuracy by around 30% in the deepest model. EAD and C&W attacks yield slightly better outcomes, indicating that these models retain minimal classification power under these perturbations. JSMA proves noticeably disruptive, dropping accuracy by more than 15% across all depths, underscoring the difficulty of achieving robust classification on high-resolution images under adversarial conditions.

The Felidae dataset shows slightly higher resistence against all perturbations in the deeper models. ResNet-18 achieves worse accuracy than the two deeper models against all attacks, usually with a difference of around 5%. The FGSM attack also seems to cause the most misclassifications, with the best-performing ResNet-101 decreasing by about 22%.

5.10 Comparing Impacts of DVBI and SVBI on Model Resilience

When comparing DVBI models with their non-DVBI ResNet counterparts under adversarial attacks, DVBI consistently demonstrates enhanced robustness across datasets and attack types. For instance, under the FGSM attack, DVBI models reliably outperform base ResNet models. On MNIST, the ResNet-18 DVBI model nearly doubles the accuracy of the base model, while for CIFAR-10, it achieves a 35% accuracy increase. Even in challenging datasets like ImageNet64, where base model performance is below 1%, DVBI models show notable improvements, with ResNet-101 achieving around 10% higher accuracy. Similarly, for attacks like EAD, C&W, and JSMA, DVBI models maintain significantly higher performance compared to base models. The improvements are particularly striking under the EAD attack, where DVBI models achieve accuracies up to 85% higher on MNIST and more than 80% higher on CIFAR-10 compared to their base counterparts. Across all tested attacks and datasets, DVBI models exhibit consistently greater resistance to adversarial perturbations, underscoring their robustness.

In contrast, comparing SVBI models to DVBI reveals nuanced performance differences. On MNIST, SVBI models consistently achieve higher accuracies across all attacks compared to DVBI. Particularly under FGSM, the SVBI ResNet-50 maintains an accuracy of more than 90%. SVBI also retains a strong advantage under EAD and C&W attacks on MNIST, with accuracies typically above 90%, outperforming DVBI. However, on CIFAR-10, DVBI models generally hold the edge, with ResNet-50 achieving higher accuracies under EAD and C&W attacks. FGSM presents a unique challenge for both methods on CIFAR-10, with SVBI slightly outperforming DVBI by about 10% on ResNet-18.

On more complex datasets like SVHN and ImageNet64, DVBI maintains a slight advantage. Under gradient-based attacks such as FGSM, SVBI models experience a steeper drop, particularly for ResNet-18 on SVHN, where performance declines by 10%. Similarly, on ImageNet64, DVBI shows higher resilience across all attacks, particularly JSMA, where SVBI trails DVBI by up to 14% on ResNet-50. On the Felidae dataset, SVBI demonstrates a marginal advantage under FGSM, achieving up to 7% higher accuracy with ResNet-101, but DVBI consistently performs better against more complex attacks like EAD and C&W. Overall, DVBI models exhibit higher robustness to adversarial attacks across most datasets and attacks. While SVBI models occasionally outperform DVBI, particularly on MNIST and under certain attack types, they generally underperform in more challenging scenarios, especially on deeper architectures and more complex datasets. These findings will be further explored in Chapter 6, with a focus on the underlying factors contributing to the performance disparities between DVBI and SVBI. The relative accuracies of the examined models are summarized in Table 5.14, in addition to the accumulated BPP values.

	Model	MNIST			CIFAR-10			SVHN			ImageNet64			Felidae		
Method		Base	DVBI	SVBI	Base	DVBI	SVBI	Base	DVBI	SVBI	Base	DVBI	SVBI	Base	DVBI	SVBI
Unperturbed %	R18	0.9893	0.9693	0.9749	0.8829	0.8784	0.8616	0.9404	0.938	0.936	0.3805	0.3724	0.3527	0.7543	0.7343	0.7229
	R50	0.9912	0.9479	0.9849	0.8921	0.8493	0.7588	0.9488	0.9412	0.9418	0.4218	0.4226	0.3659	0.7657	0.8086	0.7457
	R101	0.9875	0.9424	0.9747	0.8905	0.8421	0.8195	0.9437	0.9501	0.9255	0.4217	0.4208	0.3555	0.7914	0.7857	0.7457
Bpp	R18	24	0.014968	0.075537	24	0.065169	0.831367	24	0.007535	0.582609	24	0.005887	1.284756	24	0.000224	0.104587
	R50	24	0.012039	0.110552	24	0.012401	0.383747	24	0.00842	0.411298	24	0.013346	1.859822	24	0.000327	0.307937
	R101	24	0.021434	0.062611	24	0.015071	0.487917	24	0.009945	0.303902	24	0.015326	1.857516	24	0.000504	0.248868
FGSM %	R18	-0.5475	-0.1334	-0.0552	-0.7456	-0.3952	-0.4875	-0.6595	-0.4231	-0.5184	-0.3776	-0.2859	-0.3272	-0.7543	-0.2714	-0.2458
	R50	-0.3516	-0.1722	-0.037	-0.6885	-0.3784	-0.2472	-0.5806	-0.4209	-0.4448	-0.417	-0.3218	-0.3222	-0.7286	-0.3486	-0.2286
	R101	-0.4463	-0.1929	-0.0576	-0.6931	-0.3762	-0.3408	-0.6831	-0.4794	-0.4759	-0.4171	-0.3174	-0.3113	-0.7485	-0.3286	-0.2257
EAD %	R18	-0.9458	-0.0713	-0.0387	-0.8701	-0.0285	-0.0396	-0.9237	-0.0834	-0.0736	-0.3785	-0.0113	-0.0196	-0.7514	-0.0372	-0.0772
	R50	-0.8836	-0.1279	-0.0364	-0.8801	-0.0461	-0.0452	-0.8878	-0.1276	-0.0723	-0.4173	-0.0115	-0.0181	-0.6686	-0.0857	-0.0428
	R101	-0.9392	-0.0871	-0.0384	-0.8775	-0.0362	-0.0488	-0.9323	-0.0978	-0.0896	-0.4194	-0.0104	-0.0184	-0.7343	-0.0514	-0.0457
C&W %	R18	-0.9801	-0.0973	-0.0591	-0.8552	-0.0476	-0.0777	-0.9054	-0.102	-0.1286	-0.3543	-0.0196	-0.0548	-0.7257	-0.0343	-0.0743
	R50	-0.9827	-0.202	-0.056	-0.8514	-0.0656	-0.0542	-0.9209	-0.1367	-0.1094	-0.3737	-0.0256	-0.0552	-0.7543	-0.0943	-0.0428
	R101	-0.9864	-0.1409	-0.0531	-0.8865	-0.0572	-0.0633	-0.9432	-0.1249	-0.1185	-0.4166	-0.0253	-0.0553	-0.7828	-0.0428	-0.0486
JSMA %	R18	-0.9478	-0.3442	-0.1365	-0.8762	-0.1757	-0.2078	-0.9358	-0.2173	-0.1608	-0.3466	-0.09	-0.1707	-0.7486	-0.0429	-0.08
	R50	-0.8363	-0.3854	-0.1978	-0.851	-0.1963	-0.1787	-0.9017	-0.3022	-0.2028	-0.3932	-0.1025	-0.1918	-0.7543	-0.0857	-0.04
	R101	-0.8457	-0.3367	-0.1689	-0.8358	-0.2123	-0.1764	-0.9408	-0.3223	-0.1939	-0.3897	-0.1112	-0.1539	-0.7857	-0.0543	-0.0543

Table 5.14: A final table showing the relative robustness gains/losses of base-, DVBI and SVBI models under various adversarial perturbations.

5.11 Feature Generalization and Vulnerability to Adversarial Attacks

Our findings seem to support the conclusion that the generalization capabilities of a neural network, while desirable for achieving high performance on clean data, appear to correlate with an increased susceptibility to adversarial attacks. Specifically, we observed that models trained with higher bits-per-pixel values, indicating greater redundancy in the feature representation, were more vulnerable to adversarial perturbations. This suggests that the more a model retains and compresses intricate feature details, the more effectively adversarial inputs can exploit these retained redundancies. Using the Felidae dataset as our benchmark, we adjusted the bpp by varying the compression weight λ [39] to control the trade-off between compressibility and accuracy. Our experiments revealed a clear pattern: as the redundancy in the feature space increased with higher bpp, adversarial attacks became progressively more impactful, leading to larger accuracy drops. This result, depicted in Figures 5.1 and 5.2 aligns with the theoretical expectation that a model holding a more intricate representation of the input data is likely maintaining unnecessary, potentially vulnerable details that adversarial perturbations can target. When tested on perturbed datasets, the models showed a similar trend as uncorrupted data. The accuracy is generally kept the same until a λ of 1000, where a stark decrease is observed. We refer to the appendix for interested readers. In our testing, we attempted to reasonably minimize the information redundancy of our models while ensuring a usable classification accuracy. We have not, however, aimed at reaching the lowest possible redundancy for any given model or dataset while keeping the prediction near-lossless, as this was not relevant to the goals of this thesis. Our conclusions from these tests reinforce

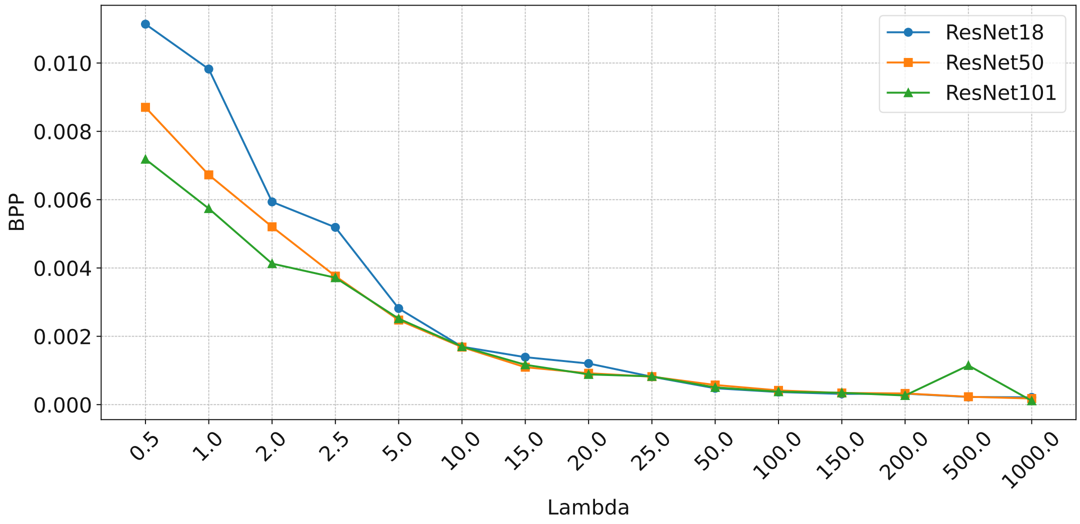


Figure 5.1: Observed information redundancy decrease (bits-per-pixel) with an increasing λ in training.

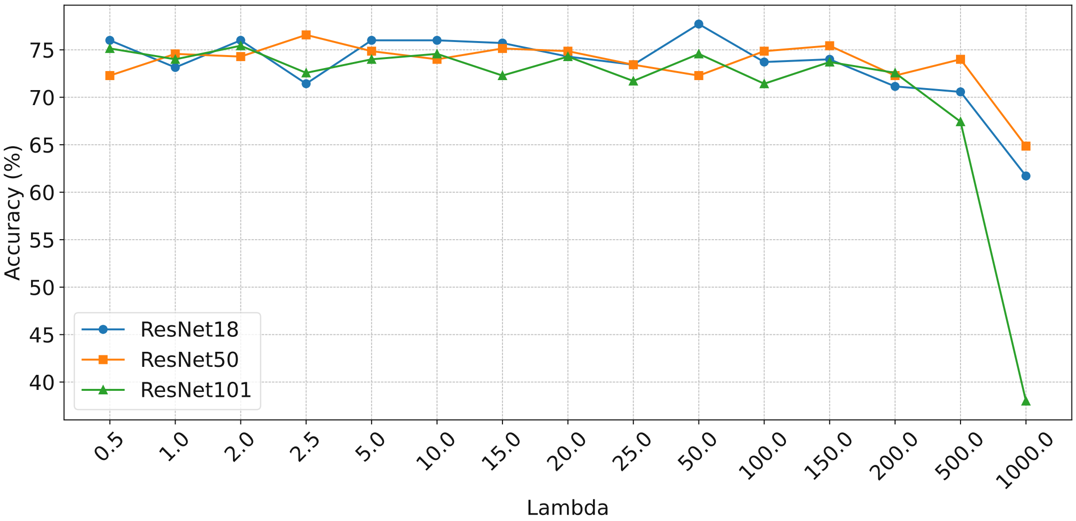


Figure 5.2: Unperturbed classification accuracy of ResNet models trained on the Felidae Dataset using different λ values (and consequently different bpp).

the notion that while low-compression (high-redundancy) feature representations improve classification accuracy, they also broaden the attack surface for adversarial exploits.

5.12 Tabacof Attack Results

The results in Table 5.15 show the Top1 accuracy of various ResNet models after applying the Tabacof attack, as well as the number of selected target labels identified by the models. The baseline ResNet models (ResNet-18, ResNet-50, and ResNet-101) achieved Top1 accuracies of 61.61%, 76.57%, and 33.17% respectively. The DVBI ResNet models showed varied results with Top1 accuracies of 52.26%, 72.66%, and 34.73%, respectively. The baseline ResNet models identified 927, 879, and 448 target labels, respectively, out of a total of 1135 target labels in the dataset. The DVBI ResNet models identified 1802, 1126, and 1641 target labels. Notably, the DVBI_ResNet-18 identified a significantly higher number of target labels, suggesting that the Tabacof attack had a pronounced effect on this model. The results indicate several key points. The DVBI_ResNet-18 model, while achieving a lower Top1 accuracy by almost 10% compared to the baseline ResNet-18, identified a disproportionately high number of target labels (1802), which is significantly higher than the total target labels in the dataset (1135). This suggests that the Tabacof attack effectively manipulated the latent space of this model, causing it to misinterpret inputs as the target labels frequently. Compared to its baseline counterpart, the significant drop in Top1 accuracy for DVBI_ResNet-18 indicates that the attack was particularly effective against the DVBI model, supporting the theory that bottleneck-injected models are more vulnerable to this type of adversarial manipulation.

Tabacof attack	Top1 Accuracy	# target labels
Base_Resnet18	61.61%	927
DVBI_Resnet18	52.26%	1802
Base_Resnet50	76.57%	879
DVBI_Resnet50	72.66%	1126
Base_Resnet101	33.17%	448
DVBI_Resnet101	34.73%	1641

Table 5.15: Tabacof attack results on different ResNet models. The column "# target labels" denotes the number of times the selected target labels were identified by the models. The total target labels in the dataset are 1135.

Similarly, the baseline ResNet-50 slightly outperformed its DVBI counterpart in terms of Top1 accuracy, achieving around 4% more compared to the DVBI_ResNet-50. Additionally, the DVBI_ResNet-50 identified more target labels (1126) than the baseline ResNet-50 (879), further suggesting that the attack affected the DVBI model more. This outcome indicates that the regularization introduced by the variational bottleneck, while beneficial in other scenarios, may actually introduce vulnerabilities when faced with attacks designed to exploit latent space manipulations, as with the Tabacof attack.

For the ResNet-101 models, the DVBI variant only slightly outperformed the baseline in Top1 accuracy, achieving only around 1% more compared. However, the DVBI_ResNet-101 identified a significantly higher number of target labels (1641) than the baseline model (448), indicating that while the accuracy difference was marginal, the DVBI model was significantly more susceptible to the Tabacof attack in terms of target label misclassification. The slight difference in accuracy seems to result from the base model being generally more susceptible to the perturbations in the input images, as indicated by the critically small number of identified target labels (448) compared to the ground truth (1135). Consequently, this would mean that while the DVBI model variant did get fooled by the attack, the inputs were too corrupted for the base model, and the accuracy ended up being comparably bad.

The analysis highlights that deeper models (ResNet-50 and ResNet-101) underperformed compared to the shallower ResNet-18 regarding Top1 accuracy and the number of identified target labels, particularly in the DVBI variants. This suggests that with their increased complexity, deeper architectures may be more prone to overfitting the adversarial perturbations introduced by the Tabacof attack, leading to poorer generalization and higher misclassification rates under these conditions.

Furthermore, the fact that the DVBI models, which incorporate a variational bottleneck, generally performed worse than their baseline counterparts under the Tabacof attack was notable. The Tabacof attack specifically targets the bottleneck-injected models by manipulating the latent space, and these results confirm that such models are indeed more vulnerable to this type of adversarial manipulation. This outcome suggests that while the variational bottleneck might help in some adversarial scenarios, it introduces significant vulnerabilities when faced with attacks designed to exploit the latent space, such as the Tabacof attack.

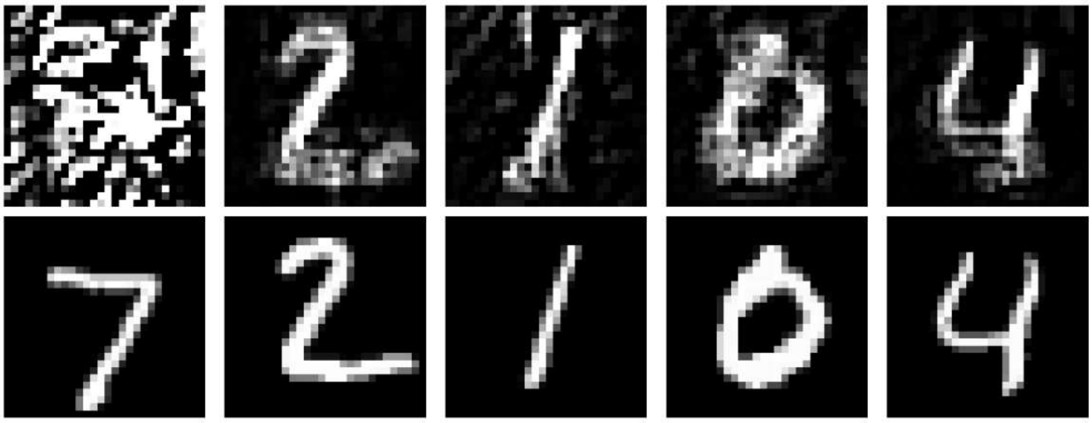


Figure 5.3: Base images (bottom) perturbed using the Tabacof method (top) against the DVBI injected ResNet-18 model with a target label of 1. Examples depicting the numbers two and four contain the most clearly visible perturbations to match this target.

CHAPTER 6

Discussion

This chapter examines our experimental findings on enhancing neural network robustness against adversarial attacks using Shallow and Deep Variational Bottleneck Injection. We compare these techniques to understand how bottleneck depth affects a model's ability to resist adversarial perturbations. We also discuss the impact of various attack methods, the role of dataset complexity and model depth, the significance of L-norm metrics, and the challenges encountered during our research, aiming to provide insights for future improvements in model resilience.

6.1 Interpretation of Experiment Results

The results of our experiments shed light on several key aspects of model robustness under adversarial attacks. First, the comparison between SVBI and DVBI models provides insight into the effectiveness of bottleneck techniques in mitigating the impact of adversarial perturbations. The experiments confirmed that SVBI models, while exhibiting improved robustness over baseline models, are still mostly outperformed by DVBI models. Specifically, DVBI models were better equipped to handle perturbed images, showing a marked reduction in misclassification rates when subjected to adversarial attacks. This finding is consistent with the hypothesis that deeper bottleneck injections help compress the feature space more effectively, discarding noise that would otherwise contribute to model confusion.

Interestingly, our experiments also revealed that the MNIST dataset displayed better resilience against adversarial perturbations compared to more complex image sets. A possible explanation for this could be that MNIST images, due to their simplicity, can be effectively compressed within shallower layers, leading to a naturally more robust representation. This phenomenon suggests that the inherent characteristics of the dataset, coupled with compression mechanisms, may play a significant role in adversarial resistance.

The core advantage of DVBI models lies in their ability to reduce the dimensionality of intermediate representations, thereby limiting the attack surface available to adversarial perturbations. The robustness of these models stems from the fact that deeper bottlenecks force the network to retain only the most salient features while discarding redundant information. By doing so, DVBI models create a more compressed and informative representation, which adversarial attacks find harder to exploit. In contrast, SVBI models, while still capable of filtering out some noise, do not compress the feature space to the same extent. This limits their ability to neutralize sophisticated perturbations, as the shallow bottleneck allows more irrelevant or adversarial features to pass through.

A potential reason for the observed lower resilience of SVBI models to adversarial attacks compared to DVBI may lie in the higher bits-per-pixel values typically exhibited by SVBI models. Higher bpp values indicate that SVBI encodes richer detail and finer-grained information about the input, which, while beneficial for capturing complex data distributions, may inadvertently make the model more sensitive to adversarial perturbations. These perturbations exploit the high-dimensional feature space and can more easily mislead a model that emphasizes detailed representation over robustness. In contrast, DVBI's relatively lower bpp values suggest a more compressed representation that might focus on salient features, thereby reducing the vulnerability to adversarial noise. This trade-off between representation richness and robustness highlights a key factor in the differing performances of SVBI and DVBI under adversarial conditions.

An important aspect highlighted by the experiments is the difference in performance between various adversarial attacks on SVBI and DVBI models. While the C&W attack, known for its minimal and optimized perturbations, was most effective against the base models, both DVBI and SVBI models showed greater resistance to this attack. We posit this to be the result of the attack being the most subtle from the selected suite, overwhelmingly reaching the lowest L-norms. These small but usually effective perturbations seem to effectively disappear in the intermediate representation of the input image due to the nature of the variational bottleneck. Consequently, for attacks like FGSM and JSMA, which make more obvious, large-scale alterations to the images, both DVBI and SVBI models were more susceptible compared to other attacks. These attacks, however, also affected the base models less severely, resulting in smaller relative accuracy gains from the DVBI and SVBI approaches. Despite these smaller improvements, the flat accuracy for DVBI and SVBI remained lower under FGSM and JSMA, highlighting a trade-off in robustness between different types of attacks.

The experiments also highlight the critical role of dataset complexity in determining model robustness. Models trained on more complex datasets, such as CIFAR-10 and ImageNet64, demonstrated greater sensitivity to adversarial perturbations, especially when the attacks targeted high-resolution images with intricate features. In contrast, simpler datasets like MNIST exhibited more consistent performance across both SVBI and DVBI models, though DVBI still maintained an edge. This further reinforces the notion that deeper bottleneck injections are more effective when the model needs to generalize across a broader range of features and when the attack surface is larger due to the complexity of the input data.

In summary, while SVBI models represent a significant improvement over baseline models in handling perturbed images, DVBI models provide a more robust defense against a wider range of adversarial attacks. The superior performance of DVBI models can be attributed to their ability to compress the feature space more effectively, thus minimizing the impact of noise and adversarial features. However, it is important to note that even DVBI models are not impervious to all attacks, particularly more crude ones like JSMA.

6.2 Classification Accuracy Decrease of Deeper Bottleneck Injected Models With Perturbed Datasets

The observation that deeper models, such as ResNet-50 and ResNet-101, generally exhibit lower robustness to perturbed inputs in bottleneck injected models can be attributed to several factors. Deeper networks, while capable of capturing more complex patterns and hierarchical features, also have a larger capacity to overfit to specific data distributions. This increased capacity can make them more susceptible to adversarial perturbations, as the subtle and carefully crafted modifications introduced by adversarial attacks can exploit the high-dimensional feature space these models operate in. Moreover, deeper models often rely on intricate combinations of features, and slight disruptions to these features can cause significant degradation in performance. In contrast, shallower models like ResNet-18, which rely on more general features, may not be as easily swayed by minor perturbations. Additionally, the higher number of parameters in deeper models might lead to more complex decision boundaries that are more easily exploited by adversarial attacks, resulting in a greater drop in accuracy when these models are exposed to adversarially perturbed inputs.

6.3 Effectiveness of Different Attacks

Building on our initial findings on base ResNet models, we conducted additional experiments to assess the effectiveness of various adversarial attacks on both SVBI and DVBI models. Our results corroborate the conclusion that C&W remains the most effective attack in terms of reducing base model accuracy. This is particularly evident in high-complexity datasets like ImageNet64, where the C&W attack reduced the accuracy of the baseline ResNet models to below 1%. In contrast, the simpler FGSM attack, while still effective, produced less severe accuracy drops. As mentioned, however, both bottleneck injection approaches struggled to handle high-intensity perturbations, such as those introduced by FGSM and JSMA. SVBI, in particular, showed that shallow bottleneck techniques do not seem to provide enough compression to neutralize these attacks effectively.

With the base models, the trend across datasets is more or less consistent. Deeper models (ResNet-101 and especially ResNet-50) exhibit slightly better resilience than their shallower counterparts, but the difference is marginal. This suggests that while increased depth provides some defense, it is not sufficient to fully mitigate the impact of sophisticated attacks.

Another noteworthy finding is the overall effectiveness of JSMA, even in the OnePixel implementation. Here, JSMA attacks, which focus on altering a minimal number of pixels, were able to produce misclassifications with fewer perturbations compared to other examined attacks. While JSMA did not reach the lowest accuracies from all used attacks, it nonetheless managed to fool the classifier enough to see a clear decrease in accuracy, thus highlighting the importance of choosing an attack that balances effectiveness and covertness.

The Tabacof attack provided valuable insights into the robustness of different ResNet models, revealing that the DVBI models, as expected, performed worse than their baseline counterparts when faced with targeted perturbations. The results indicate that the variational bottleneck, while beneficial in some scenarios, made the DVBI models more susceptible to the Tabacof attack, particularly regarding misclassifying a significantly higher number of target labels. This suggests that the Tabacof attack effectively exploits the vulnerabilities introduced by the variational bottleneck, manipulating the latent space to degrade model performance. Therefore, while the variational bottleneck may offer advantages in other contexts, it appears to introduce specific weaknesses that adversarial techniques like the Tabacof attack can exploit. These findings highlight the need for further refinement of DVBI models to enhance their robustness against such targeted adversarial strategies.

Despite multiple attempts, the SVBI models did not yield usable results under the Tabacof attack. This may be attributed to the unique nature of shallow bottleneck injection, which places the variational compression layer closer to low-level features rather than abstract, high-level representations. As a result, shallow bottleneck-injected models might lack the sensitivity to adversarial perturbations crafted to target complex latent spaces, as in the Tabacof method. In contrast, the DVBI models, which incorporate the bottleneck at deeper levels, demonstrated clear vulnerability to the Tabacof attack, as evidenced by their high target label misclassification rates. This contrast between DVBI and SVBI models supports the hypothesis that the Tabacof attack specifically exploits latent space manipulation and that deeper injection points amplify susceptibility to such attacks. Therefore, the statistics from the DVBI models alone are sufficient to illustrate the efficacy of the Tabacof attack as a threat to some variational bottleneck-injected architectures, underlining the need for caution in applying these compression techniques in adversarially sensitive environments.

6.4 Importance of L-Norms

In our analysis, using L_0, L_2, and L_∞ norms provided an understanding of how different types of perturbations affected model performance. L_0 norms, which count the number of pixels altered by an attack, revealed the sparsity of perturbations required to mislead models. For example, the L_0 norm for the C&W attack on MNIST was 141.6 pixels for ResNet-18, indicating that only a small number of pixel changes were necessary to reduce accuracy to 4.35%. In contrast, for CIFAR-10, the L_0 norm was significantly higher (2827 pixels), reflecting the greater complexity of this dataset and the corresponding need for more widespread perturbations to achieve misclassification.

The L_2 norm, measuring the overall magnitude of perturbations, was particularly useful in understanding how distributed attacks like FGSM impacted model accuracy. Higher L_2 norms generally correlated with greater performance degradation, especially in more complex datasets. For example, in SVHN, the L_2 norm for FGSM was 1.73 (ResNet-18), and the corresponding accuracy dropped to 28.09%. This suggests that even moderate perturbations can significantly impair model performance, particularly in datasets with intricate features.

Finally, the L_∞ norm, which tracks the maximum change to any single pixel, highlighted the vulnerability of models to localized, high-intensity perturbations. For instance, in the ImageNet64 dataset, JSMA attacks with a relatively low L_∞ norm still caused substantial accuracy drops, underscoring the impact of focused, pixel-level changes on model predictions.

By evaluating all three norms in conjunction with accuracy metrics, we gained a detailed picture of model robustness. L_0 norms provided insight into the sparsity of the attacks, L_2 norms captured the overall distortion, and L_∞ norms revealed the worst-case pixel-level changes. This multi-faceted approach allowed us to understand better the strengths and weaknesses of both SVBI and DVBI models under different adversarial conditions.

6.5 Effects of Subjectively Imperceptible Noise

While in our chosen attack methods, each technically introduces perceptible noise (objective), as indicated by our calculated L_∞ norms, some perturbations remain subtle enough not to alert a human observer (subjective).

Relating this to our research on DNN image classifiers under adversarial attacks, the effectiveness of these perturbations—whether perceptible or imperceptible—plays a crucial role in evaluating the robustness of the classifiers. Adversarial attacks often aim to generate imperceptible perturbations that cause misclassification without alerting the human observer. In this context, we have previously theorized that implementing a variational information bottleneck, either in a deep or shallow architecture, could increase the robustness of DNNs by filtering out noise that does not contribute meaningfully to the classification task. Our findings suggest that by compressing the input representation to retain only the most informative features, a variational information bottleneck seems to help mitigate the impact of both perceptible and imperceptible adversarial perturbations, leading to improved classifier performance on perturbed datasets. Notably, while our tested perturbations were all objectively perceptible, our statistics indicate that the stronger, more obvious, or more crude the perturbation, the smaller the accuracy gains from the injected bottleneck. Therefore, it would stand to reason that these models would indeed be more resilient towards imperceptible perturbations.

6.6 Limitations and Challenges

We have encountered several challenges and problems throughout this project, which we would like to outline in the hopes of aiding future researchers that might come across similar issues. The most important problem we faced was the lack of sufficient computational power for the tasks we set out to do. Analyzing the behavior of nine different model combinations each had to be trained on five distinct datasets meant our setup was effectively running non-stop for almost three months. This estimate includes iterating on training parameters to find the best balance of accuracy and speed, as well as creating adversarial perturbations using our four selected attack methods. For large-scale, broad analyses like ours, we recommend using multiple machines with more powerful GPUs, as it was often the case that even small errors in our methodology caused us to have to re-train whole models, unable to proceed with our work until this was done. A rather large problem we encountered was our inconsistent de-/normalization of the datasets. After generating all of the perturbed datasets targeting each base model, we figured out that we had supplied the attacks with normalized datasets, leading to unusable final images after the de-normalization step. While the classification accuracies did not change much, these images were not comparable to the original ones, and they were producing nonsense L-norm values. For these reasons, all attacks had to be re-applied and the perturbed datasets re-generated.

On a related note, we quickly ran into the issue of the JSMA attack not scaling well for images larger than SVHN or CIFAR (32x32). Our solution to this problem involved creating a custom JSMAOnePixel attack, as described in section 4. While this approach solved the error of the attack running out of memory, there was still a noticeable decrease in performance compared to the standard torchvision attacks, as it took almost a week to generate the perturbed images based on the ImageNet64 dataset. A faster experimental setup would have helped us with this problem as well. However, even this adjusted method would be too slow to be used on the entire ImageNet dataset, for example, as the sheer volume of test images combined with the number of iterations needed to achieve reliable results on the comparably high-resolution inputs would be too high.

In Chapter 1, we predicted that our selected attacks might produce too similar perturbations, leading to similar classification accuracies and, therefore, sub-optimal analysis results. This was not the case, and the evasion methods produced mostly distinct perturbations and, consequently, distinct results for our evaluation metrics. More details can be found in section 5.

CHAPTER 7

Conclusion and Future Work

7.1 Summary of Findings

This thesis investigates the potential of variational bottleneck injection to enhance the robustness of deep neural networks against adversarial evasion attacks. Through a series of experiments comparing models trained with Shallow Variational Bottleneck Injection, Deep Variational Bottleneck Injection, and traditional DNN architectures without bottleneck layers, we explored the resilience of these architectures to state-of-the-art adversarial attacks. The attacks used in our evaluations include FGSM, EAD, C&W, and JSMA, representing a diverse spectrum of perturbation techniques. Our findings reveal the following key insights:

- **Bottleneck Placement and Robustness:** Both SVBI and DVBI models demonstrated improved robustness against adversarial attacks compared to the base models. While DVBI exhibited the highest resilience, SVBI also outperformed the base models across all attack scenarios. The success of bottleneck techniques, particularly DVBI, suggests that limiting the information flow in deeper layers allows the network to focus on essential features, thereby reducing the impact of adversarial perturbations.

- **Impact of Network Depth:** Contrary to expectations, in both SVBI and DVBI setups, deeper models (e.g., ResNet-50 and ResNet-101) often showed worse performance on perturbed data compared to shallower models like ResNet-18. This was specifically the case when dealing with lower-resolution datasets and contrasting the base models, where deeper architectures outperformed ResNet-18 more often than not. Higher-resolution datasets like ImageNet64 and Felidae showed the opposite effect. This finding suggests that while injecting bottlenecks may reduce the benefits of network depth with lower-resolution datasets, depth continues to provide an advantage with higher-resolution datasets. This may occur because deeper networks are better equipped to capture fine-grained details and complex patterns present in higher-resolution images, whereas bottleneck injection can overly constrain the information flow needed to process lower-resolution, perturbed data effectively.

- **Effectiveness Against Different Attack Types:** The models trained with variational bottleneck techniques demonstrated varying degrees of robustness against different types of adversarial attacks. One of the most striking results was the improved resilience of both DVBI and SVBI models to the C&W attack, which was the most effective against the base models. The subtle nature of the C&W perturbations, designed to evade detection by manipulating image features without introducing obvious distortions, seemed to be countered more effectively by the bottleneck-injected models. This contrasts with more overt attacks like FGSM and JSMA, where the base models were more susceptible. The Tabacof attack targetting the autoencoder process directly also proved to be effective against our DVBI models, indicating that while the examined bottleneck methods show clear improvements in robustness against standard evasion attacks, they may also introduce potential additional attack vectors.

- **Generalization to Non-Adversarial Perturbations:** In addition to adversarial attacks, we examined the models' performance on common image corruptions. The bottleneck-injected models, particularly DVBI, showed improved generalization and robustness to these non-adversarial perturbations compared to traditional architectures. This finding suggests that bottleneck injection not only aids in defending against adversarial attacks but also improves overall model resilience to a wider range of input disturbances.

In summary, our results highlight the potential of variational bottleneck techniques, particularly DVBI, as a promising defense mechanism against adversarial attacks. While the inclusion of shallow bottlenecks (SVBI) offers clear computational, and robustness advantages, its slightly greater susceptibility to adversarial manipulations necessitates further refinement if it is to be considered as a potential defensive method against these attacks. The unexpected performance degradation in deeper models with bottleneck injection raises important considerations for network design, and the enhanced resistance to C&W-like attacks suggests that bottleneck techniques may be especially valuable in defending against more subtle forms of adversarial manipulation. Ultimately, our research contributes to a deeper understanding of how bottleneck injection influences DNN security and opens avenues for future work aimed at optimizing these techniques for real-world applications.

7.2 Contributions to the Field

In addition to the experimental findings, this thesis makes several notable contributions to the field of adversarial machine learning, particularly in the realm of defensive strategies for deep neural networks. First and foremost, we have expanded the understanding of how variational information bottleneck techniques affect DNN resilience against adversarial attacks. Our comprehensive evaluation of Shallow Variational Bottleneck Injection and Deep Variational Bottleneck Injection offers important insights into their respective strengths and limitations. These findings confirm the utility of both methods for enhancing the robustness of machine learning models, with DVBI demonstrating superior protection across a variety of attacks and datasets. This clarity adds an important layer of understanding to the broader discussion of adversarial robustness, helping to position future research in this area better.

Another contribution lies in the introduction of a novel experimental methodology that evaluates DNN robustness using multiple attack strategies across different datasets. By integrating a diverse set of adversarial attack methods—including gradient-based attacks such as FGSM and optimization-based attacks like C&W—alongside image corruption tests, we developed a replicable framework for future studies. This methodology allows for a thorough examination of the effectiveness of defensive techniques under a variety of conditions and provides an empirical basis for comparing the resilience of different neural architectures and defensive mechanisms. This experimental approach is valuable not only for adversarial machine learning but also for research seeking to strengthen model security against other forms of data tampering or input distortion.

Moreover, this work contributes to the growing body of literature that applies bottleneck techniques to enhance DNN security. While previous studies primarily focused on bottleneck techniques for tasks like feature compression or generalization improvement, our research emphasizes their defensive potential. By demonstrating that both the SVBI and DVBI methods can protect image classification models against sophisticated adversarial perturbations, this thesis expands the practical applications of VIB techniques, showing that they are not only tools for improving model efficiency but also for increasing adversarial robustness. This perspective opens up new avenues for the deployment of these techniques in security-sensitive applications, such as autonomous driving or medical diagnostics, where model reliability is crucial.

7.3 Recommendations for Further Work

While our results support the effectiveness of VIB techniques, several areas remain open for further research. A promising direction for further research involves developing hybrid bottleneck techniques that combine the benefits of both SVBI and DVBI. Such an approach could provide a more balanced trade-off between computational efficiency and robustness, particularly in environments where resources are constrained. Hybrid techniques could leverage the computational and deployment simplicity of SVBI, while harnessing the deeper feature compression and adversarial resilience that DVBI offers in more complex architectures.

In addition to hybrid approaches, expanding the application of VIB techniques beyond the ResNet architecture could provide further evidence for our claims. While our work focused on the performance of SVBI and DVBI in ResNet-based models, it remains an open question whether these techniques would be equally effective in other types of deep learning architectures, such as transformers. A cross-architectural study of VIB methods would help determine the generalizability of these techniques and could potentially reveal new insights into their adaptability across different model structures and tasks.

Another promising area for further research may be applying VIB techniques in transfer learning scenarios. Many modern machine learning models are pre-trained on large datasets and then fine-tuned for specific tasks. Understanding how VIB affects transferability, and how adversarial robustness holds up when models are adapted for new tasks, could lead to more effective strategies for safeguarding pre-trained models from adversarial attacks. Given that transfer learning is widely used in real-world applications, developing methods to secure these models against adversarial manipulation could have substantial practical implications.

Future work should also include more experiments to investigate the effects of the Tabacof attack on SVBI models, as the present study did not fully explore this aspect. While initial attempts with SVBI models yielded limited results, further testing could reveal whether Tabacof's latent space manipulation techniques affect shallow bottleneck architectures under different conditions or parameter settings. Understanding the SVBI models' potential vulnerabilities or resilience to such adversarial attacks would provide a complete picture of how variational bottleneck placements influence model robustness.

Lastly, there is room for further refinement and improvement of VIB techniques, particularly in making them more computationally efficient. While DVBI demonstrates significant robustness improvements, it also comes with a computational cost, especially in deeper models. Future research could focus on optimizing the trade-offs between bottleneck size, information retention, and adversarial resilience. Attacks aimed specifically at the variational bottleneck also present a not insignificant danger as we have proven via the Tabacof method. Exploring new ways to make DVBI or similar techniques more lightweight, efficient and secure without sacrificing their defensive benefits would make them more practical for real-world deployment.

In conclusion, while this thesis provides strong evidence supporting the effectiveness of VIB techniques—particularly DVBI—there is still considerable scope for further exploration. Future research should aim to make these methods more efficient, expand their applicability to a broader range of architectures, and develop hybrid approaches that better balance performance with computational cost. By addressing these challenges, the field can continue to advance toward more robust and secure neural networks capable of withstanding the evolving landscape of adversarial threats.

APPENDIX A

Images Perturbed Using the Selected Attacks

A.1 FGSM Perturbed Example Images

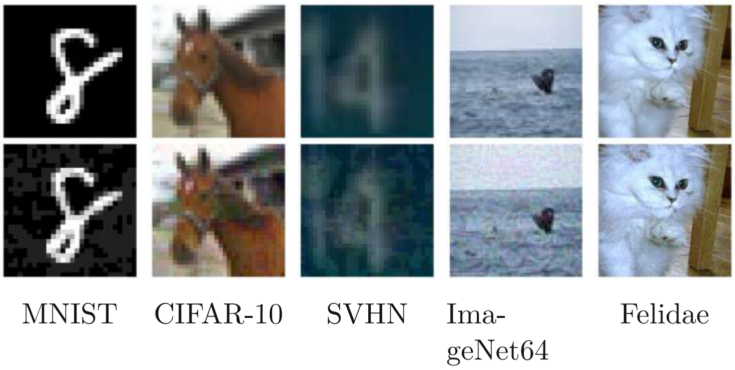


Figure A.1: FGSM image perturbations targetting the ResNet-18 model.

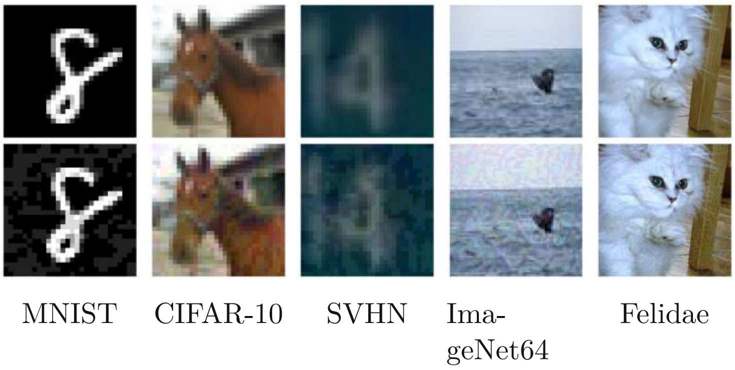


Figure A.2: FGSM image perturbations targetting the ResNet-50 model.

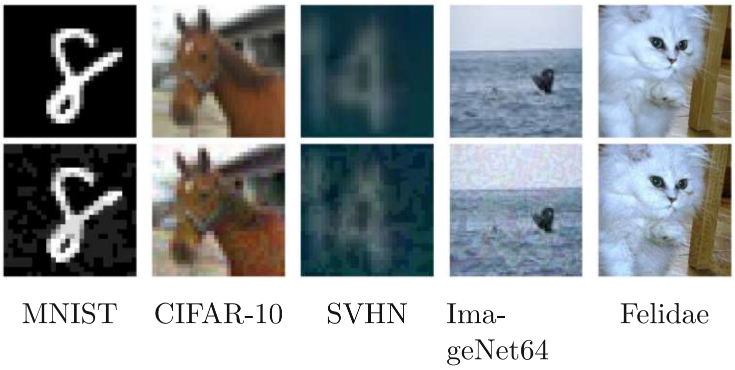
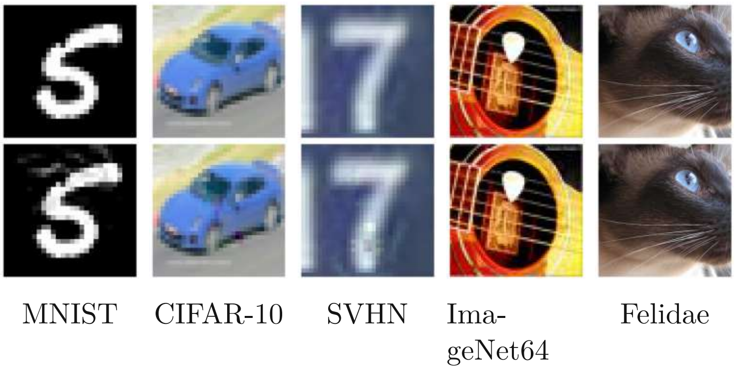


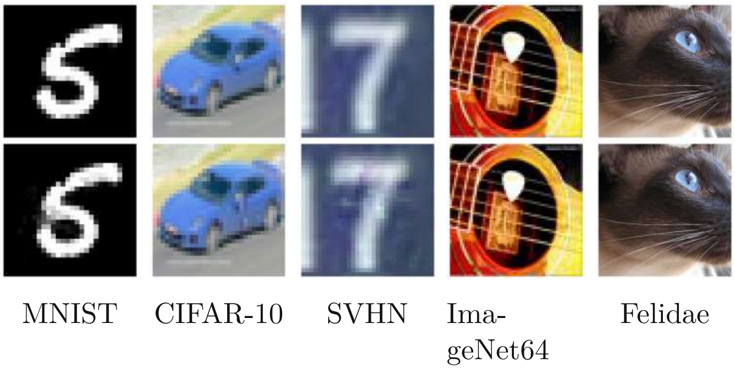
Figure A.3: FGSM image perturbations targetting the ResNet-101 model.

A.2 EAD Perturbed Example Images



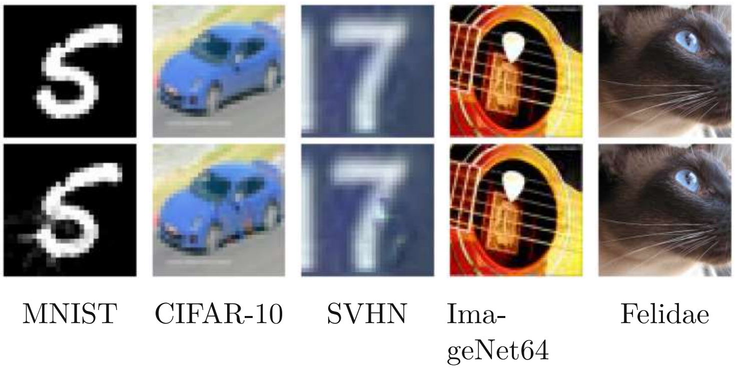
MNIST CIFAR-10 SVHN ImageNet64 Felidae

Figure A.4: EAD image perturbations targetting the ResNet-18 model.



MNIST CIFAR-10 SVHN ImageNet64 Felidae

Figure A.5: EAD image perturbations targetting the ResNet-50 model.



MNIST CIFAR-10 SVHN ImageNet64 Felidae

Figure A.6: EAD image perturbations targetting the ResNet-101 model.

A.3 C&W Perturbed Example Images

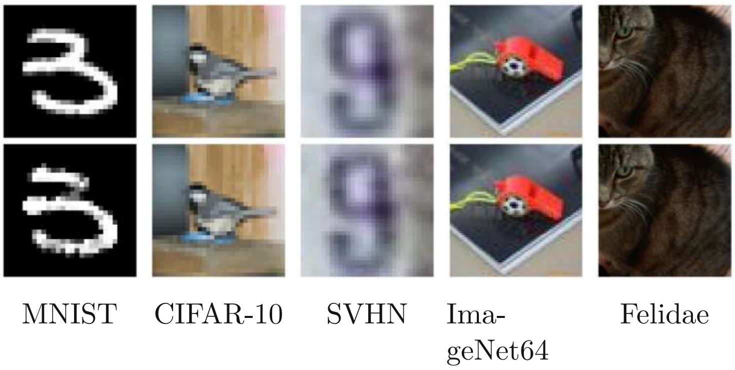


Figure A.7: C&W image perturbations targetting the ResNet-18 model.

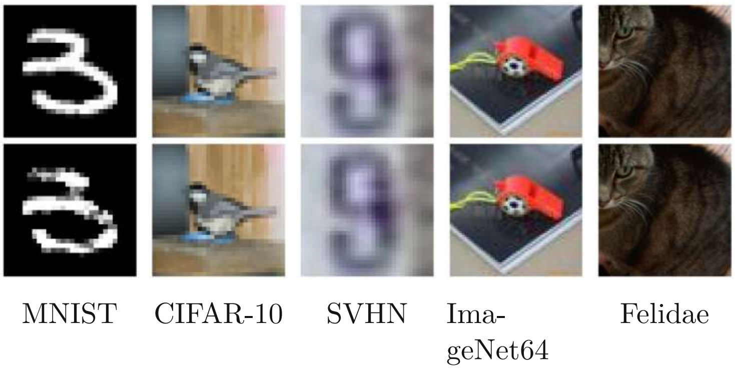


Figure A.8: C&W image perturbations targetting the ResNet-50 model.

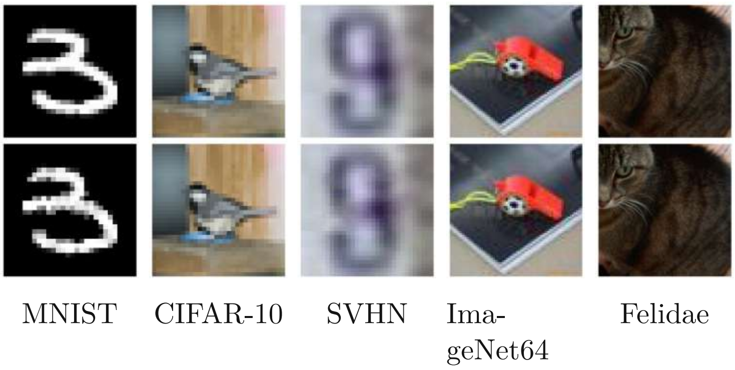
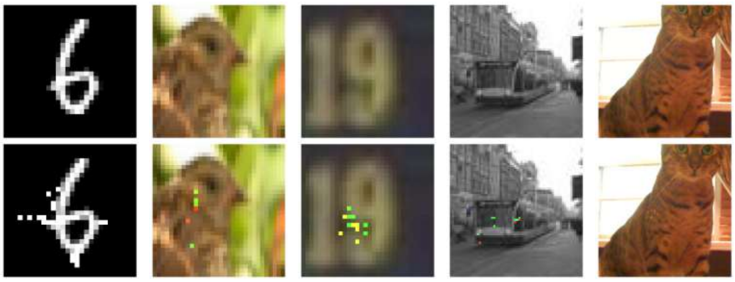


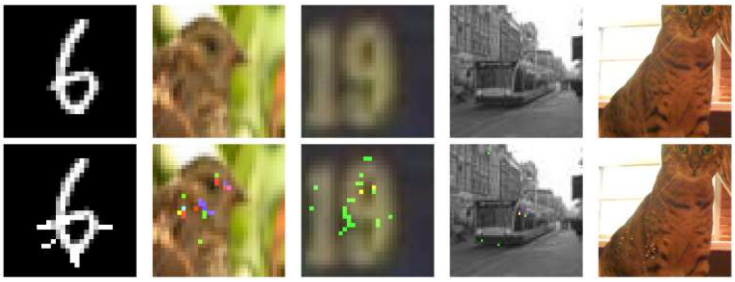
Figure A.9: C&W image perturbations targetting the ResNet-101 model.

A.4 JSMA Perturbed Example Images



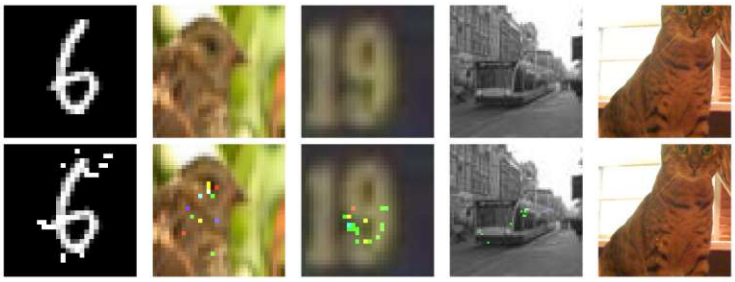
MNIST CIFAR-10 SVHN ImageNet64 Felidae

Figure A.10: JSMA image perturbations targetting the ResNet-18 model.



MNIST CIFAR-10 SVHN ImageNet64 Felidae

Figure A.11: JSMA image perturbations targetting the ResNet-50 model.



MNIST CIFAR-10 SVHN ImageNet64 Felidae

Figure A.12: JSMA image perturbations targetting the ResNet-101 model.

APPENDIX B

Accuracies on Perturbed Images Related to BPP

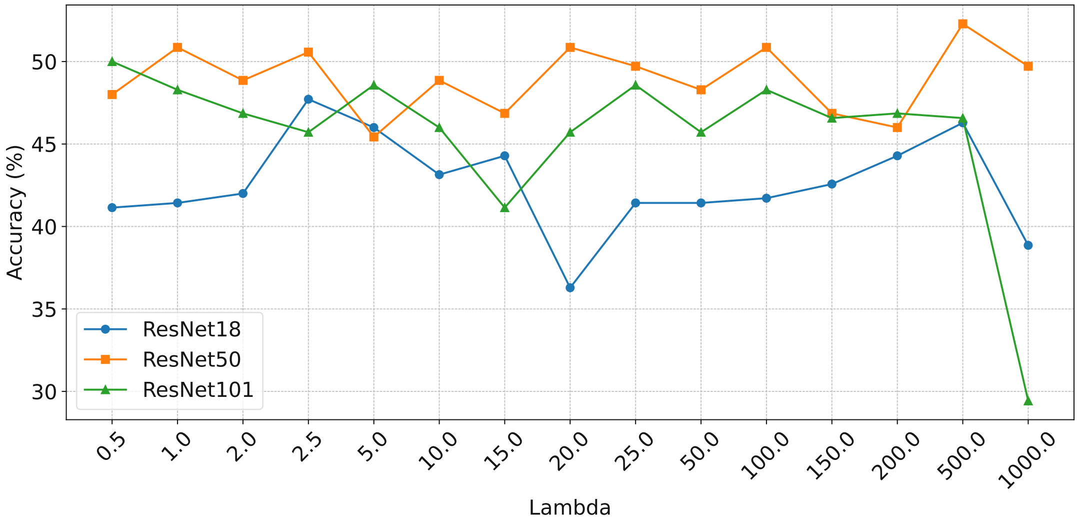


Figure B.1: FGSM perturbed classification accuracy of ResNet models trained on the Felidae Dataset using different λ values (and consequently different bpp).

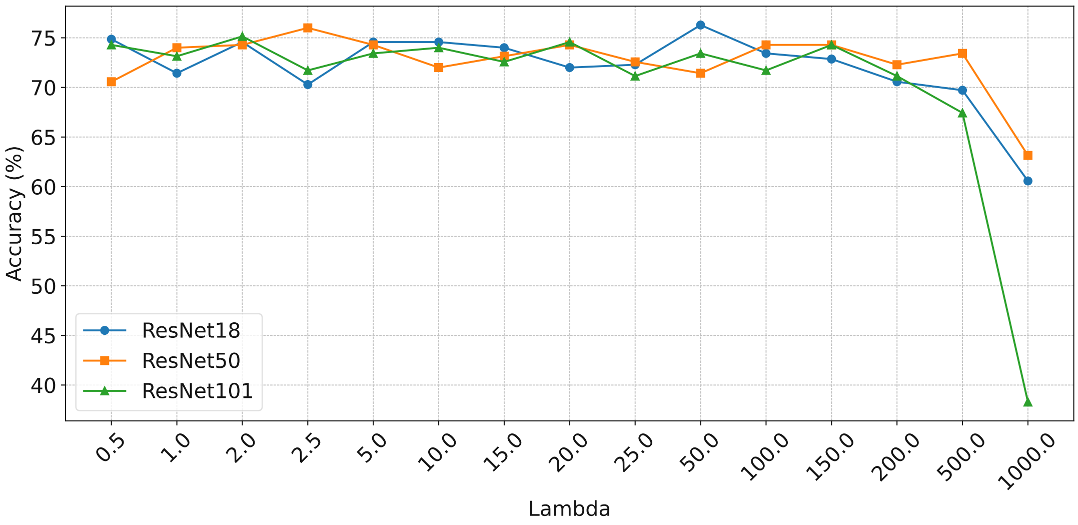


Figure B.2: EAD perturbed classification accuracy of ResNet models trained on the Felidae Dataset using different λ values (and consequently different bpp).

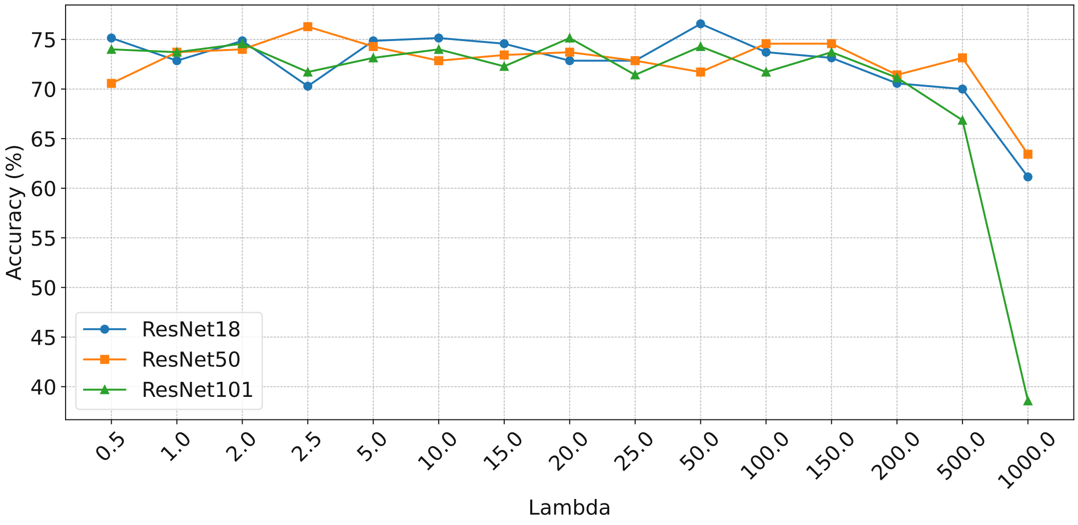


Figure B.3: C&W classification accuracy of ResNet models trained on the Felidae Dataset using different λ values (and consequently different bpp).

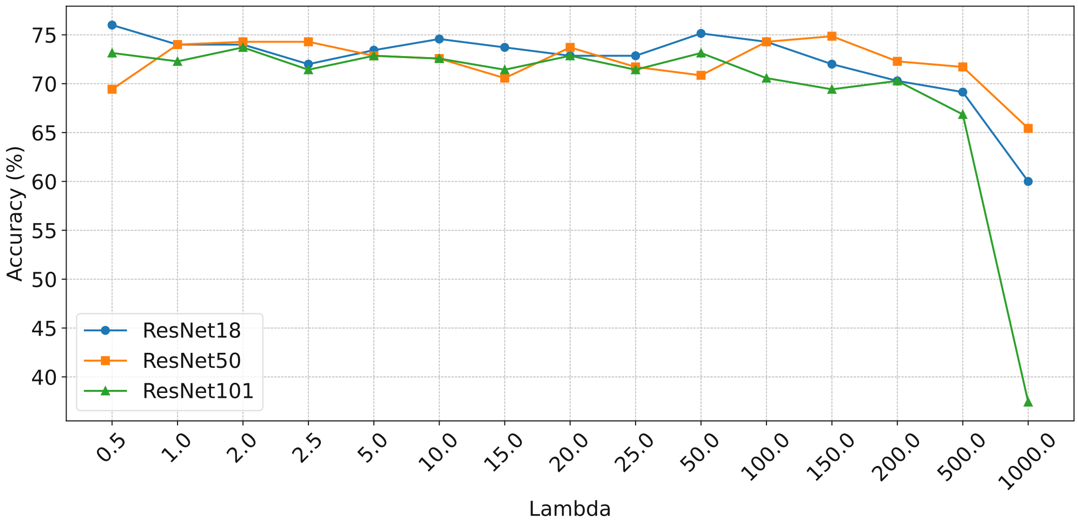


Figure B.4: JSMA classification accuracy of ResNet models trained on the Felidae Dataset using different λ values (and consequently different bpp).

Overview of Generative AI Tools Used

During the writing process, I utilized the AI model GPT-4, provided by ChatGPT [28], in a limited capacity. Its use was restricted to idea generation, assistance with LaTeX, and verifying the proper formulation of text according to scientific standards. I did not use this tool for literature research due to its unreliability. To minimize potential errors or inaccurate responses, I directly uploaded the sources to be used as files. Any unfamiliar facts received from the tool were verified against the uploaded sources and appropriately cited according to standard academic practices.

Another helpful tool in the writing process was Semantic Scholar [35], a platform that employs AI-driven technologies to identify relevant scientific articles. Features such as intelligent filters, automatic summaries, and citation analyses significantly supported the research process. The platform enabled me to quickly find relevant studies based on keywords, fields of study, or citation networks, thereby improving the quality and efficiency of the literature review.

Additionally, I used the application Grammarly [14] to check spelling and grammar. Since Grammarly also offers sentence rephrasing features, I mention it here as a generative AI application. However, I did not use it for explicitly generating or rewriting text based on inputs.

Übersicht verwendeter Hilfsmittel

Während des Schreibprozesses habe ich das KI-Modell GPT-4, bereitgestellt durch ChatGPT [28], in begrenztem Umfang eingesetzt. Die Nutzung erfolgte ausschließlich zur Anregung von Ideen, Unterstützung bei der Arbeit mit LaTeX und Überprüfung der korrekten Formulierung des Textes laut wissenschaftlichen Standards. Für die Recherche von Literatur habe ich dieses Tool wegen Unzuverlässigkeit nicht verwendet. Um mögliche Fehlerquellen oder fehlerhafte Antworten zu minimieren, habe ich die zu verwendenden Quellen direkt als Datei hochgeladen. Mir nicht vertraute Fakten, die ich von diesem Tool erhalten habe, habe ich mit den hochgeladenen Quellen überprüft und gemäß den üblichen akademischen Standards korrekt gekennzeichnet.

Ein weiteres hilfreiches Tool im Schreibprozess war Semantic Scholar [35], eine Plattform, die KI-gestützte Technologien einsetzt, um relevante wissenschaftliche Artikel zu identifizieren. Durch Funktionen wie intelligente Filter, automatische Zusammenfassungen und Zitieranalysen unterstützte Semantic Scholar den Rechercheprozess erheblich. Dabei ermöglichte mir die Plattform, relevante Studien basierend auf Schlüsselbegriffen, Themenfeldern oder Zitationsnetzwerken schnell zu finden, was die Qualität und Effizienz der Literaturrecherche deutlich verbesserte.

Zusätzlich habe ich die Anwendung Grammarly [14] zur Überprüfung von Rechtschreibung und Grammatik eingesetzt. Da Grammarly auch Funktionen zur Umformulierung von Sätzen bietet, erwähne ich sie hier als generative KI-Anwendung. Das explizite automatische Erstellen oder Umschreiben von Text basierend auf Eingaben habe ich nicht verwendet.

List of Figures

List of Tables

Bibliography

[1] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[2] M. F. Alzantot, Y. Sharma, S. Chakraborty, and M. B. Srivastava. Genattack: practical black-box attacks with gradient-free optimization. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018.

[3] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[4] N. Carlini and D. Wagner. Defensive distillation is not robust to adversarial examples, 2016.

[5] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2016.

[6] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. *ArXiv*, abs/1709.04114, 2017.

[7] T. Chen and Z. Ma. Towards robust neural image compression: Adversarial attack and model finetuning. *ArXiv*, 2021.

[8] T. Chen and Z. Ma. Toward robust neural image compression: Adversarial attack and model finetuning. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7842–7856, Dec. 2023.

[9] Y. Dubois, B. Bloem-Reddy, and C. J. Maddison. Lossy compression for lossless prediction. *ArXiv*, abs/2106.10800, 2021.

[10] A. Furutanpey, P. Raith, and S. Dustdar. Frankensplit: Efficient neural feature compression with shallow variational bottleneck injection for mobile edge computing. *IEEE Transactions on Mobile Computing*, 2023.

[11] A. Furutanpey, Q. Zhang, P. Raith, T. Pfandzelter, S. Wang, and S. Dustdar. Fool: Addressing the downlink bottleneck in satellite computing with neural feature compression. *ArXiv*, abs/2403.16677, 2024.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples, 2015.

[13] G. Goswami, A. Agarwal, N. K. Ratha, R. Singh, and M. Vatsa. Detecting and mitigating adversarial perturbations for robust face recognition. *International Journal of Computer Vision*, 127:719 – 742, 2019.

[14] Grammarly. Grammarly, 2024. Accessed: 2024-11-17.

[15] T. Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019.

[16] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae. Beyond transmitting bits: Context, semantics, and task-oriented communications. *IEEE Journal on Selected Areas in Communications*, 41(1):5–41, 2022.

[17] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations, 2018.

[18] D. Hendrycks and T. G. Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv: Learning*, 2018.

[19] F. Khalid, M. A. Hanif, S. Rehman, R. Ahmed, and M. A. Shafique. Trisec: Training data-unaware imperceptible security attacks on deep neural networks. *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 188–193, 2018.

[20] H. Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

[21] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

[22] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2016.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017.

[24] T. maintainers and contributors. Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision`, 2016.

[25] Y. Matsubara, M. Levorato, and F. Restuccia. Split computing and early exiting for deep learning applications: Survey and research challenges. *ACM Comput. Surv.*, 55(5), dec 2022.

[26] Y. Matsubara, R. Yang, M. Levorato, and S. Mandt. Supervised compression for resource-constrained edge computing systems. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Jan. 2022.

[27] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[28] OpenAI. Chatgpt, 2024. Accessed: 2024-11-17.

[29] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387, 2015.

[30] N. Papernot, P. Mcdaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, 2015.

[31] PapersWithCode. Paperswithcode. `https://paperswithcode.com/`. Accessed: 2024-07-29.

[32] Pedro Tabacof and Julia Tavares and Eduardo Valle. adv_vae. `https://github.com/tabacof/adv_vae`. Accessed: 2024-08-06.

[33] PyTorch. torchvision.datasets. `https://pytorch.org/vision/stable/datasets.html`. Accessed: 2024-08-24.

[34] PyTorch. torchvision.models. `https://pytorch.org/vision/0.9/models.html`. Accessed: 2024-08-24.

[35] S. Scholar. Semantic scholar, 2024. Accessed: 2024-11-17.

[36] A. Shamir, O. Melamed, and O. BenShmuel. The dimpled manifold model of adversarial examples in machine learning. *ArXiv*, abs/2106.10151, 2021.

[37] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models, 2017.

[38] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[39] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ball'e, A. Shrivastava, and G. Toderici. End-to-end learning of compressible features. *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3349–3353, 2020.

[40] K. Song. Adversarial.js. `https://kennysong.github.io/adversarial.js/`. Accessed: 2024-07-29.

[41] Statista. Number of artificial intelligence (ai) tool users globally from 2020 to 2030 (in millions) [graph]. `https://www.statista.com/forecasts/1449844/ai-tool-users-worldwide`. Accessed: 2024-08-24.

[42] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks, 2014.

[43] P. Tabacof, J. Tavares, and E. Valle. Adversarial images for variational autoencoders. *ArXiv*, abs/1612.00155, 2016.

[44] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method, 2000.

[45] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[46] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle, 2015.

[47] C. Xie, Y. Wu, L. van der Maaten, A. Yuille, and K. He. Feature denoising for improving adversarial robustness, 2019.