ACIT5900

Master's thesis

In

Applied Computer and Information Technology (ACIT)

August 2024

Applied Artificial Intelligence

Evaluating the Robustness of Deep Learning Models Against Adversarial Attacks

---

Quang Dung Martin Phan

Department of Computer Science

Oslo Metropolitan University, Technology Art and Design

1 Abstract

With the rapid integration of Artificial Intelligence (AI) into various sectors, contemporary technological landscapes have been revolutionized. This has notably enhanced capabilities in Deep Learning models, which are used in a wide range of applications, from automated vehicles to security surveillance. Nevertheless, the vulnerability of these systems to adversarial attacks, in which inputs are cleverly manipulated to mislead AI models, presents a substantial risk to their dependability and safety. This master's thesis explores the durability of object detection models when faced with adversarial attacks, with the goal of connecting theoretical research to real-world application robustness.

The research primarily focuses on evaluating deep learning models, in adversarial contexts created using techniques like the Fast Gradient Sign Method (FGSM). These models were selected based on their widespread usage and demonstrated success in handling intricate image datasets, such as CIFAR-10, which served as the main dataset for training and testing the models. This dataset offers a wide variety of images for thorough testing of the models' ability to accurately recognize and classify distorted data.

With the goal of gaining a deep understanding, the thesis examines various challenging situations to evaluate how well the model performs and to discover ways to enhance its ability to withstand adversarial attacks. The study utilizes a well-organized approach that includes a thorough examination of existing literature to situate the research within the current academic and practical contexts of AI security. The empirical analysis included thorough model training methods, utilizing both traditional and adversarial training techniques to evaluate the effectiveness of various training modifications.

This research adds to the ongoing discussion on improving AI security, providing valuable strategies to strengthen DL models against the ever-changing risk of adversarial attacks. Through deepening our comprehension of these vulnerabilities and defenses, the study contributes to the creation of AI applications that are more robust, guaranteeing their dependability and trustworthiness in crucial real-world operations.

2 Preface

This master's thesis, titled "Evaluating the Robustness of Deep Learning Models Against Adversarial Attacks in Applied Artificial Intelligence," represents the culmination of my Master's education in Applied Computer and Information Technology, with a specialization in Applied Artificial Intelligence at OsloMet. The planning phase began in late 2023, and the thesis was researched and composed from January to August 2024.

Throughout the course of this project, I have significantly enhanced and abilities in my area of expertise. Exploring this subject has expanded my knowledge of research methods, such as conducting systematic literature reviews and utilizing advanced model training techniques to address adversarial threats. Before this thesis, my knowledge of adversarial attacks in artificial intelligence was limited. This research project has greatly expanded my understanding and practical skills, equipping me for future professional and academic endeavors.

I am incredibly grateful to my academic advisor, Professor Lothar Fritsch, for their exceptional guidance and invaluable feedback during the research process. His extensive expertise in machine learning and cybersecurity greatly enhanced this project. I am grateful to [Industry Partner's Name] for their partnership, which offered valuable industry perspectives that greatly influenced this research. Lastly, I want to express my deep gratitude to my friends and family. Their unwavering support and patience were invaluable to me throughout my studies, particularly during the demanding phase of this thesis work.

3 Table of Contents

4 List of Figures

5 List of *Tables*

Chapter I

1 Introduction

The emergence of artificial intelligence (AI) has been a pivotal moment in the history of technology, with its uses affecting nearly every facet of contemporary life. Artificial intelligence (AI) is having a significant and widespread impact on everything from personalised recommendations on streaming services to driverless cars traversing city streets and sophisticated algorithms making important healthcare choices. AI has been incorporated into daily tasks, which has improved human capacities and created a new paradigm for how people and machines interact. But as AI applications grow more and more ingrained in our daily lives, maintaining their dependability and security against malevolent interference has become a crucial concern. Adversarial attacks represent a serious risk to the reliability and integrity of these systems because they can cause subtle, frequently undetectable changes to input data that could lead to AI models being misled. (Goodfellow et al., 2014)

Investigating adversarial threats in practical settings is not only a research endeavor but also an urgent need to protect the AI-powered aspects of our everyday existence. AI applications in the actual world are subject to a wide range of inputs, unlike controlled environments, which makes them vulnerable to deliberately constructed adversarial examples. These attacks might have far-reaching effects, such as making self-driving cars misread traffic signs, fooling security checkpoint facial recognition software, or tricking medical imaging software into making false diagnoses. It is crucial to create strong AI systems that can resist these difficulties because such hostile manipulations are a possibility. This project aims to bridge the gap between laboratory conditions and the complexities of the external environment by extending research on adversarial attacks from theoretical models to real-world applications. This will provide insights into the resilience of AI systems against adversarial threats in their natural operational contexts.

Therefore, the goal of this master's thesis research is to investigate adversarial attacks in theory and in practical real-world scenarios while highlighting the importance of striking a

balance between the advancement of AI technologies and protecting them from new threats. It recognizes how AI applications are transforming society and fostering breakthroughs that have the potential to completely reshape the future. Confronting these systems' inherent vulnerabilities, it offers a thorough analysis of adversarial attack strategies and their consequences for practical AI applications. The project seeks to add significant knowledge to the ongoing discussion on AI security by thoroughly examining this dynamic topic, providing tactics to protect AI applications from malicious vulnerabilities, and guaranteeing their safe incorporation into daily life.

1.1 Project Description

This thesis provides an in-depth investigation of adversarial attacks in object detection via an organized composition that includes an empirical analysis, a methodological explanation, and a critical evaluation of the literature. The review of the literature searches large databases, such as Scopus, using a rigorous selection procedure to guarantee the inclusion of peer-reviewed articles that support experiments and findings in the field of adversarial machine learning, with a focus on image and video data types.

The thesis is divided into smaller parts, which can be formulated as such:

1. **Investigate common types of Adversarial attacks and Deep Learning Models.**
    a. **Literature Review:** Conduct systematic review using databases such as Scopus to identify peer-reviewed articles focusing on adversarial machine learning, specifically targeting object detection with image and video data types.
    b. **Establish Inclusion and Exclusion Criteria**: Define criteria to select relevant studies, focusing on those providing empirical results and employing recognized adversarial contexts.
2. **Research on building a machine learning environment.**
    a. **Coding Framework**: Utilize Python and TensorFlow, chosen for their extensive libraries and compatibility with GPU computations.
    b. **Environment Setup**: Set up a coding environment using Python, TensorFlow, and the necessary GPU support software (CUDA toolkit, cuDNN).

3. **Build and implement machine learning models for object detection.**
    a. **Model Selection and Training**: Employ models such as Convolutional Neural Networks (CNNs), MobileNetV2, and ResNet50, focusing on their training and evaluation under adversarial conditions.
    b. **Model Evaluation**: Test these models to identify vulnerabilities and assess their robustness against adversarial attacks.

1.2 Project Goal

My main goal in this thesis is to delve into the complexities of adversarial attacks on DL models and evaluate the ability of modern detection models to withstand these threats. The research seeks to measure the impact of adversarial attacks on object detection and suggest defensive strategies to improve system resilience. This will be accomplished by conducting thorough literature reviews and conducting experimental studies. An evaluation will be conducted to assess the vulnerability of different detection models, such as Convolutional Neural Networks (CNNs), MobileNetV2, and ResNet50. This evaluation aims to provide a comprehensive understanding of the impact of various adversarial strategies on diverse datasets. The study aims to establish effective strategies for creating and improving image classification models that can withstand malicious attacks, thus making valuable contributions to the field of applied artificial intelligence. This guidance will be essential for developing AI systems with enhanced security. Here is the outlined goal of the study:

Goal: Assess the resilience of Deep learning Models against adversarial attacks.

- Examine the current literature and perform experimental research on various models to gain insights into the weaknesses of deep learning models when it comes to adversarial attacks.
- Examine how adversarial attacks can affect the accuracy of deep learning models.

1.3 Research Questions

This research explores machine learning techniques for detecting objects in the presence of adversarial conditions. It especially examines how adversarial attacks might be designed to deceive deep learning systems and evaluates the resilience of these systems against such attacks. The study utilizes the CIFAR10 dataset to train and deep learning models, such as Convolutional Neural Networks (CNN), MobileNetV2, and ResNet50. The models undergo analysis in different hostile settings to detect flaws and improve their robustness.

Research Questions

- Which deep learning model is the most effective in preserving accuracy and reliability when faced with adversarial attacks in object detection?
- How does the integration of adversarial training affect the ability of object detection models to withstand aggressive examples?
- What is the impact on the performance of deep learning when they are trained and tested on augmented data from the CIFAR10 dataset to enhance their ability to withstand adversarial attacks?

The purpose of these questions is to enhance the comprehension of adversarial robustness in DL models, hence aiding in the construction of more secure AI applications.

1.4 Project Outline

Chapter I

The first chapter provides an overview of artificial intelligence (AI) in general, with a particular emphasis on adversarial attacks and how they affect deep learning models. The introduction sets the stage for the research by emphasizing the necessity of strengthening the resilience of AI systems against adversarial threats. It also highlights the significance of comprehending and mitigating adversarial risks in real-world applications, such as autonomous driving and security systems.

Chapter II

An extensive review of artificial intelligence is given in the background part, with an emphasis on object detection. It describes the history and many branches of artificial intelligence while highlighting how relevant they are to the problems presented by hostile

threats. This chapter also surveys the body of work on adversarial attacks in the field of object detection, emphasizing prior discoveries and pointing out gaps that the present study attempts to fill.

Chapter III

This chapter provides a detailed description of the research methods used. It goes over the methodical process for doing a literature review, including the selection criteria for pertinent research and the databases that are searched. The experimental setup, including the coding frameworks, tools, and the dataset are also explained in length in this chapter. Additionally, the choice of object identification models, including CNN, MobileNetV2, and ResNet50, for assessing their resistance to adversarial manipulation, is covered.

Chapter IV

The results of the literature review and the empirical investigation are presented in this chapter. A summary of the literature is provided, highlighting recurring themes and significant flaws in the state of object detection defenses at the moment. The evaluation and training outcomes of the models are also provided in detail, demonstrating how well the various models functioned in hostile environments and evaluating the potency of alternative defensive tactics.

Chapter V

The results are interpreted in the discussion chapter and are contrasted with extant hypotheses and earlier studies. It investigates the findings' practical ramifications, namely how they might be used to improve the security and dependability of AI systems in real-world situations and advance the subject of AI security.

Chapter VI

The research is summed up in the last chapter, which highlights the contributions made to the field of adversarial machine learning and the security implications for AI applications. It also suggests areas for additional research based on the study's findings and open-ended questions. These represent possible directions for future research.

1.5 Limitations

This thesis will not encompass a comparison of all pre-trained models available for transfer learning due to several constraints. Specifically, only a selection of pre-trained

models compatible with the existing hardware and software environments used in this study will be considered. Compatibility issues arise mainly from variations in the underlying architecture requirements and the computational intensity of certain models, which may not be supported efficiently by the available infrastructure. Furthermore, although it would be ideal to evaluate the performance of these models across a broader range of datasets to ensure generalizability and robustness of findings, this approach was not feasible. Limitations in hardware resources and time constraints inherent in a one-semester thesis project necessitated focusing only on selected datasets that are most relevant and manageable within the given period. This focused approach ensures depth rather than breadth in analysing the performance of chosen models under specific conditions, while acknowledging the broader applicative potential in future work.

Chapter II

2 Background

- This section I will introduce and define the key concepts and terminologies used throughout the thesis.

2.1 Artificial Intelligence

Artificial Intelligence (AI) encompasses the vast field of developing intelligent machines that can carry out tasks typically associated with human intelligence. AI encompasses a wide range of technologies and approaches, spanning from robotics to reasoning. Its goal is to improve machine perception, learning, and decision-making. AI systems can vary from basic algorithms that solve specific tasks to advanced machine learning and deep learning systems that continuously learn and adjust. AI has a wide range of applications across various domains. It can recognize speech, interpret complex data, automate operational tasks, and solve critical business challenges. (Team, 2023)

Machine Learning

Machine Learning (ML) is a branch of AI that involves the use of algorithms and statistical models to enable computers to carry out tasks without being explicitly programmed, instead relying on patterns and inference. This is how AI achieves its intelligence - by learning and improving automatically through experience. Machine learning algorithms construct a mathematical model using sample data, referred to as "training data," to make predictions or decisions without the need for explicit programming.

Supervised Learning

Supervised learning involves training a model using a dataset that has been labelled. Each piece of the training data is matched with the correct output, ensuring accuracy. The objective of supervised learning is to train the model to make precise predictions when provided with new input. Supervised learning is commonly employed in various applications where past data can be used to anticipate future events. Examples include fraud detection in banking, email filtering, and predicting customer behavior. (Delua, 2021)

Unsupervised Learning

Unsupervised learning entails training a model with unclassified and unlabeled information, enabling the algorithm to operate independently without any guidance. Here, the machine's objective is to categorize unstructured data based on similarities, patterns, and differences, without any prior data training. Unlike supervised learning, there is no teacher provided and the machine is not given any training. Thus, the machine is limited to discovering the concealed patterns in unlabeled data independently. Unsupervised learning is often applied to transactional data, such as identifying customer segments in marketing data or detecting anomalies that may indicate fraud in transaction data.

2.1.1 Deep Learning

Deep Learning (DL) is a specialized form of Machine Learning that utilizes deep neural networks to effectively capture intricate patterns and make informed decisions. It emulates the human brain's capacity to combine various layers of information to achieve greater levels of comprehension and abstraction. Deep learning algorithms require a significant amount of data to train the models effectively, enabling them to make decisions with remarkable speed and accuracy, often surpassing human capabilities. These networks utilize multiple layers of non-linear processing units, allowing them to effectively learn and represent intricate relationships within the data. Deep learning has brought about a significant transformation in various domains, including image and speech recognition, natural language processing, and autonomous vehicle systems. Its remarkable capability to automatically extract features without human involvement has greatly enhanced accuracy in tasks like object detection and sentiment analysis.(Alzubaidi et al., 2021)

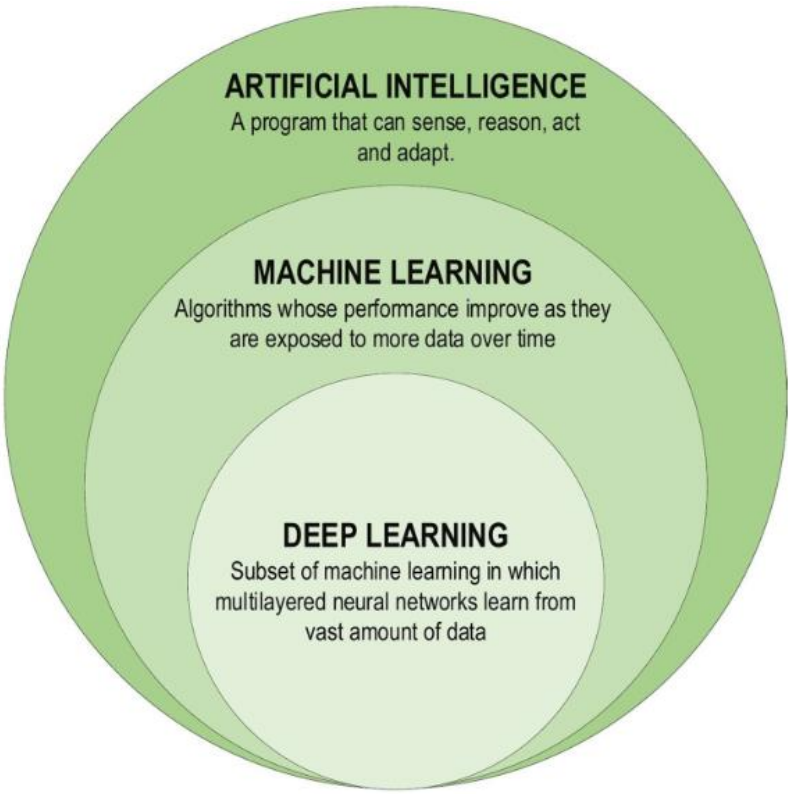

Figure 1: Connection between AI, ML and DL. The image is from a research paper by Alzubaidi et al., 2021.

Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a class of deep neural networks highly effective in areas such as image recognition and classification. CNNs are specifically designed to process pixel data and are structured similarly to the connectivity pattern of neurons in the human brain, particularly the visual cortex. CNNs use a hierarchy of layers to process input features, including convolutional layers that filter inputs for useful information, pooling layers that reduce dimensionality, and fully connected layers that determine the output from the feature analysis. The major advantage of CNNs is their ability to develop an internal representation of a two-dimensional image, allowing them to capture spatial hierarchies in data. This architecture makes CNNs exceptionally efficient in handling image and video data, outperforming standard deep learning models that lack convolutional and pooling layers.

Benefits of CNNs over Traditional Machine Learning algorithms

Deep Learning (DL) is a specialized form of Machine Learning that utilizes deep neural networks to effectively capture intricate patterns and make informed decisions. It emulates

the human brain's capacity to combine various layers of information to achieve greater levels of comprehension and abstraction. Deep learning algorithms require a significant amount of data to train the models effectively, enabling them to make decisions with remarkable speed and accuracy, often surpassing human capabilities. These networks utilize multiple layers of non-linear processing units, allowing them to effectively learn and represent intricate relationships within the data. Deep learning has brought about a significant transformation in various domains, including image and speech recognition, natural language processing, and autonomous vehicle systems. Its remarkable capability to automatically extract features without human involvement has greatly enhanced accuracy in tasks like object detection and sentiment analysis.

2.2 Overview on Deep Learning in Object Detection

Introducing Deep Learning in Computer Vision

Deep learning has had a significant impact on computer vision, revolutionizing the field and leading to advancements in various challenging tasks. These tasks involve various aspects of neural networks, such as image classification, object detection, semantic segmentation, and adversarial robustness. Image classification is an essential task that involves organizing images into specific categories. It serves as a crucial step before tackling more complex challenges. An interesting application can be found in the field of medical imaging, where it plays a crucial role in helping doctors diagnose diseases based on image data (Zhou, 2022).

Advances in Object Detection

Expanding upon the basics of image classification, object detection takes it a step further by not only categorizing but also pinpointing multiple objects within an image. This dual functionality greatly improves the usefulness of object detection in real-world situations like autonomous driving and surveillance systems. Object detection algorithms have undergone significant advancements, progressing from basic frameworks like R-CNN to more refined and efficient architectures such as Fast R-CNN and Faster R-CNN. Zhou provides a comprehensive review on the latest deep learning advances in object detection (Zhou, 2022).

Semantic Segmentation

Semantic segmentation enhances the capabilities of computer vision systems by categorizing each pixel of an image into predefined classes. This is essential for achieving a thorough understanding of a scene, which is necessary for applications such as autonomous navigation and detailed environmental mapping.

Addressing Adversarial Robustness

Adversarial robustness in neural networks is a crucial field that deals with the vulnerabilities of these models to adversarial attacks. These attacks involve making slight but purposeful modifications to inputs in order to confuse the model. In a study conducted by Madry and his team, they explored ways to improve the resilience of neural networks against adversarial attacks. Their goal was to create models that can withstand or correct these perturbations, ultimately making neural networks more reliable for security-sensitive applications (Madry et al., 2017).

Deep Learning Object Detection Architectures.

Deep Learning models usually break down the task into two main components: localization and classification. Two-stage detectors such as Faster R-CNN rely on the region proposal network (RPN) to predict object boundaries using anchor boxes. The RPN also assigns objectness scores to these regions, providing an indication of the likelihood of an object being present. The proposed regions are classified by another network component to determine the object categories, including identifying irrelevant proposals as "background" (Zhou, 2022).

Nevertheless, although two-stage detectors attain impressive accuracy, their speed is often compromised, rendering them less ideal for real-time applications. On the other hand, single-stage detectors like YOLO and SSD combine localization and classification in a single step, prioritizing speed and efficiency rather than absolute accuracy. The balance between accuracy and performance is crucial when choosing object detection models for real-time applications that require quick decision-making (Zhou, 2022).

Overall, the progress of deep learning applications in computer vision—from basic image classification to more intricate tasks such as object detection and adversarial robustness—demonstrates the continuous advancements and persistent obstacles in this

ever-changing field. As these technologies advance, they hold the potential to unleash even greater capabilities in different sectors.

2.3 Adversarial Attacks and Defenses in Deep Learning

Adversarial Attacks

Adversarial attacks take advantage of the weaknesses found in machine learning models, especially deep neural networks (DNNs), to manipulate them into making incorrect decisions. These attacks involve creating input data that may seem normal to human observers, but ultimately results in incorrect model outputs. The paper from Goodfellow, Shlens, and Szegedy (2015) highlights the vulnerability of these models to perturbations due to their linear characteristics in high-dimensional spaces. Methods such as the Fast Gradient Sign Method (FGSM) are introduced to efficiently generate adversarial examples by leveraging the model's gradients to maximize loss (Goodfellow et al., 2014).

In their study, Ren et al. (2020) provide a detailed categorization of attacks according to the attacker's level of knowledge about the model. They distinguish between white-box attacks, where the attacker possesses full knowledge of the model, gray-box attacks, where the attacker has only partial knowledge, and black-box attacks, where the attacker has no knowledge of the model's internal workings. These attacks demonstrate how model vulnerability can be exploited, highlighting the potential for manipulation of models across different systems(Ren et al., 2020).

Adversarial Defenses

As a response to these vulnerabilities, adversarial defenses strive to enhance the resilience of deep learning models against such attacks. In their study, Goodfellow et al. (2015) propose the use of adversarial training as a means to enhance the robustness of models. This approach involves training models with both clean data and adversarial examples. This approach is recognized as a supplementary method of regularization, which has the potential to improve the model's ability to generalize (Goodfellow et al., 2014).

Ren et al. (2020) explore a wider array of defensive strategies, encompassing both heuristic and certified defenses. Heuristic defenses, such as adversarial training with methods like Projected Gradient Descent (PGD), have shown to be effective in practice, although they do not come with theoretical guarantees. However, certified defenses offer

mathematical guarantees of resilience against specific attack models, although they may not perform as effectively in real-world scenarios compared to heuristic methods (Ren et al., 2020).

Correlation between Attacks and Defenses

The interplay between adversarial attacks and defenses underscores an ongoing competition in the realm of machine learning security. Advancements in attack methodologies drive the evolution of defensive strategies, creating a continuous cycle of improvement. This interaction not only contributes to technological advancements but also enhances our comprehension of the core characteristics of deep learning models, including their vulnerability to adversarial examples and the efficacy of various defense mechanisms. Both articles highlight the importance of continuous research in order to create stronger and more secure AI systems that can withstand constantly changing adversarial tactics.

2.4 Selecting the Dataset

A dataset in the field of machine learning and artificial intelligence is a curated collection of data that is meticulously prepared and utilized for the purpose of training and assessing machine learning models. Just like a software engineer, a dataset is crucial as it provides the initial input for models to learn and then be tested for accuracy, reliability, and resilience against different challenges. The performance of these models is greatly influenced by the quality, relevance, and diversity of the data in a dataset. Datasets come in all shapes and sizes, from small and straightforward collections of data to enormous pools of intricate information spanning various domains.

CIFAR-10 is a well-known dataset that is widely used in the field of machine learning, specifically for evaluating and comparing different machine learning algorithms in the area of image recognition. The dataset contains a total of 60,000 images. These images are 32x32 in size and are in color. They are evenly distributed across ten different classes, which consist of various objects such as cats, dogs, birds, and other items that are commonly used for testing object recognition algorithms. With its diverse range of data and large size, CIFAR-10 is an ideal choice for conducting thorough experiments. It provides a great opportunity to evaluate the performance of deep neural networks in both normal and adversarial scenarios.

This attribute is essential when contemplating the necessity of subjecting models to a range of situations that replicate real-life obstacles.

The CIFAR-10 dataset is highly valuable for studying adversarial attacks and defenses in deep learning models because of its diverse image content and structured nature. With its widespread use in the academic community, the dataset serves as a shared platform for evaluating the efficacy of various adversarial techniques and defense mechanisms. For example, techniques such as the Fast Gradient Sign Method (FGSM) can be used to test the impact of perturbations on model accuracy in a controlled and realistic way using CIFAR-10's diverse image set. Furthermore, the standardized structure and complexity of CIFAR-10 provides an ideal environment for thorough experimentation and improvement of defenses against adversarial attacks. Techniques like adversarial training and Projected Gradient Descent (PGD) can be extensively tested and refined, enabling researchers to create stronger models that can effectively counter potential real-world vulnerabilities. The decision to focus on CIFAR-10 for this research endeavor was driven by the aim to improve model security and reliability in the face of adversarial threats.

2.5 Artificial Neural Network

Artificial Neural Networks (ANNs) are computational models that draw inspiration from the human brain. They are designed to recognize patterns by simulating the interconnected network of neurons found in a biological brain. ANNs are composed of nodes or artificial neurons that are interconnected by links, allowing signals to be transmitted between neurons. The neurons function by processing inputs, which are then weighted and directed through a function that determines the signal's path and strength within the network. This configuration enables ANNs to execute intricate calculations quickly, making them well-suited for a wide range of tasks, including speech recognition and weather prediction. Essentially, ANNs acquire the ability to perform tasks by analyzing examples, typically without the need for task-specific programming (LeCun et al., 2015).

2.5.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) excel at handling data with a grid-like structure, like images. Convolutional Neural Networks (CNNs) utilize a mathematical

operation known as convolution. This operation, unlike general matrix multiplication, allows for the sharing of learning parameters throughout the network. This sharing feature is crucial for CNNs' efficient handling of the high dimensionality of raw images, resulting in a significant reduction in the number of parameters that require training. Typically, CNN architectures consist of three layers. The first layer filters inputs to extract useful information, the second layer reduces dimensionality while keeping the most important information, and the third layer determines the output based on the features extracted by the previous layers (Krizhevsky et al., 2012).

The practical application of CNNs in the field of image recognition is widely known, demonstrating their superior performance compared to earlier models on difficult datasets. As seen in the ImageNet challenge, CNNs have significantly reduced error rates for object recognition, highlighting their ability to automatically extract hierarchical features from training data. This feature makes them particularly well-suited for tasks that benefit from the ability to handle local translation of input features, such as facial recognition or autonomous driving (Russakovsky et al., 2015).

2.5.2 Transfer Learning

Transfer learning is an incredibly effective technique in machine learning that allows us to apply knowledge gained from solving one problem to a different, yet related problem. It's like leveraging previous experience to tackle new challenges. This approach has become increasingly popular in the realm of deep learning because it allows for the utilization of pre-trained models on extensive datasets. These models can then be adjusted to excel in specific tasks using smaller datasets. For example, models trained on the extensive ImageNet dataset have been effectively applied to various tasks, such as medical image diagnostics and object recognition in autonomous vehicles. Transfer learning has the ability to speed up training and enhance model performance when data is limited. This is particularly beneficial in situations where data collection is costly or privacy constraints restrict the amount of available data (Pan & Yang, 2010).

There are numerous benefits to utilizing transfer learning. First and foremost, it significantly decreases the requirement for extensive amounts of labeled data, which can often pose a significant challenge when training advanced deep learning models. With the

help of pre-trained networks and extensive datasets, practitioners can achieve impressive accuracy even with limited data points. They can further enhance the model's performance by fine-tuning it for their specific tasks. Additionally, transfer learning can enhance the speed of learning and performance of models, which is essential for deploying them in real-world applications with limited computational resources and time constraints. This is especially important in fields such as healthcare, where making fast and precise predictions can literally save lives. Transferring learned features across different tasks enhances the versatility and accessibility of deep learning models, benefiting a wide range of users and industries.

2.5.3 Pre-trained Models

Pre-trained models available through the Keras library represent a significant advancement in the field of deep learning, particularly for image classification tasks. These models are based on neural networks that have been previously trained on large and diverse datasets, such as ImageNet. They encapsulate a deep understanding of image features, from basic textures to complex objects, making them highly effective for a variety of visual recognition tasks.

Understanding Pre-trained Models

Pre-trained models are a valuable resource for developers and researchers. Instead of starting from scratch, one can utilize these pre-trained models that have already learned powerful and versatile feature representations. Well-known models such as VGG16, ResNet50, InceptionV3, and MobileNet are extensively utilized and differ in terms of architecture, performance, and computational efficiency, making them suitable for various image classification tasks.

Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3

Figure 2: Available models through keras. From Keras Documentation (Keras, n.d.-a)

Benefits for Image Classification

1. **Enhanced Accuracy:** Pre-trained models have been found to enhance the accuracy of image classification systems. These models offer advanced feature-extracting capabilities that are typically difficult to achieve when training a model from scratch with limited data. Research indicates that models that have been pre-trained on datasets such as ImageNet can be further optimized to achieve higher levels of accuracy when applied to smaller datasets (He et al., 2015).
2. **Development Efficiency:** Using pre-trained models can greatly speed up the development process of image classification systems. This is because the foundational network architectures and weights have already been fine-tuned and tested across different tasks.
3. **Reduced Computational Demand:** Training deep neural networks requires significant computational resources. Pre-trained models are great for applications with limited computational resources because they require less computational power and time for training. In fact, only the final layers might need to be adapted for a new task, saving you a lot of time and effort.

4. **Versatility through Transfer Learning:** Pre-trained models are incredibly flexible due to the concept of transfer learning. This requires modifying a model created for one task to suit a different yet similar task. By making adjustments to only the final layers, pre-trained models can be easily tailored to suit new applications (He et al., 2015).

Practical Applications

When it comes to putting a pre-trained model from Keras into action for a fresh image classification task, the usual approach is to load the model without its top layer (the original output classification layer), and then incorporate new layers that are tailored to the specific task at hand. This process, called fine-tuning, adjusts the model to work with new categories that were not part of the original dataset.

For instance, a model such as ResNet50, originally trained on ImageNet, can be fine-tuned to accurately classify various types of vehicles. While the foundational layers that extract general features are preserved, the classification layers are customized to identify new, specific categories.

To summarize, pre-trained models are essential tools in machine learning that improve the efficiency, accuracy, and practicality of creating reliable image classification systems. They have played a crucial role in pushing the field forward, serving as a solid foundation for numerous applications (He et al., 2015).

2.5.4 MobileNetV2

The MobileNetV2 architecture is a highly efficient deep learning model specifically designed for mobile and edge device applications. It builds upon the original MobileNet by utilizing inverted residual blocks with bottleneck features, making it even more effective. This design modification greatly decreases the number of parameters compared to its predecessor, resulting in improved efficiency without compromising performance. MobileNetV2 is designed to accommodate a range of input sizes, with a suggestion to use sizes larger than 32x32 pixels. It is worth noting that using larger image sizes often results in improved performance (Keras, n.d.-b).

Key Features of MobileNetV2

1. **Inverted Residuals and Linear Bottlenecks:** These structural enhancements contribute to maximizing the computational efficiency of the model. The inverted residuals facilitate seamless transmission of information and gradients across the network, which is vital for effectively training deep networks.
2. **Versatility in Input Size:** For optimal performance, it is common to use an input size of 224x224 pixels, although the model is capable of handling larger inputs. This adaptability makes it appropriate for various image sizes and uses. In this thesis 128x128 was utilized because of memory constraint.
3. **Modifiable Width via Alpha Parameter:** The 'alpha' parameter can be used to modify the width of the network, enabling users to adjust the number of filters according to the available computational resources or the desired complexity of the application. This feature allows MobileNetV2 to be flexible and suitable for various situations.
4. **Optional Top Layer:** Users have the option to include or exclude the top layer, depending on whether they want to use the network for feature extraction or train it end-to-end for a specific classification task.
5. **Pre-trained Weights:** Users have the option to include or exclude the top layer, depending on whether they want to use the network for feature extraction or train it end-to-end for a specific classification task.
6. **Transfer Learning and Fine-Tuning:** This model is perfect for transfer learning and fine-tuning situations. It allows you to adapt the pre-trained MobileNetV2 to new tasks by retraining the top layers.
7. **Customizable Features for Advanced Usage:** It offers choices for the type of pooling to utilize (average or max pooling when the top is excluded), as well as the number of classes, which is essential for customizing the model to include new categories beyond the 1,000 classes of ImageNet.

Preprocessing and Integration:

To ensure optimal utilization, it is crucial to preprocess input images by employing the keras.applications.mobilenet_v2.preprocess_input function. This function scales input pixels within the range of -1 to 1, aligning them with the model's training data. In addition,

the model can be initialized with a specific input tensor for more advanced applications where inputs are shared across multiple networks.

Practical Applications

The MobileNetV2 model is well-suited for mobile and edge devices because of its low computational demand. Additionally, it consistently delivers strong performance across a wide range of image classification tasks. The ability to fine-tune it makes it a top choice for developers who want to quickly and efficiently deploy high-performance **models.**

2.5.5 ResNet50

The ResNet50 architecture is a widely recognized model in the field of deep learning for image recognition, renowned for its utilization of deep residual networks. First presented in the influential paper "Deep Residual Learning for Image Recognition" at CVPR 2015, ResNet50 has emerged as a fundamental tool in the advancement of deep neural networks, enabling them to excel in intricate visual recognition assignments (Keras, n.d.-c).

Key Features of ResNet50

1. **Deep Residual Learning**: The fundamental concept behind ResNet50 involves the incorporation of residual learning blocks. These blocks solve the problem of vanishing gradients, which is a common issue in deep networks, by incorporating shortcut connections that bypass one or more layers. These shortcuts enable the smooth flow of gradients through the network, enabling the training of deeper networks to enhance performance without adding complexity.
2. **Configuration Flexibility**: ResNet50 can be customized based on specific needs:
    - include_top: Specifies if the fully-connected layer should be included in the network, allowing the model to be used for both classification and feature extraction purposes.
    - weights: Provides choices for initializing the model weights, either randomly or by loading pre-trained weights from ImageNet, making it easy to use right away or customize for specific applications.
    - input_shape: Enables the use of input images of different sizes, offering flexibility for a wide range of deployment situations.

- **Preprocessing Requirements**: Inputs need to go through specific preprocessing steps in order to be ready for processing by ResNet50. The preprocess_input function in keras.applications.resnet converts RGB images to BGR and then centers each color channel with respect to the ImageNet dataset norms, without any additional scaling. This ensures that the input data is standardized to align with the training environment of the model.

3. **Pooling Options**: When not including the top layer, ResNet50 allows for different pooling mechanisms:
    - None: Outputs a 4D tensor from the last convolutional block.
    - avg: Applies global average pooling, resulting in a 2D tensor.
    - max: Applies global maximum pooling.
4. **Transfer Learning and Fine-Tuning**: ResNet50 is a fantastic option for transfer learning, thanks to its powerful and versatile feature extraction capabilities. Researchers and developers can easily customize the model for new tasks by adjusting the upper layers while keeping the deeper layers unchanged.

ResNet50 is especially well-suited for demanding image classification tasks where achieving high model accuracy is crucial. Its utilization of residual blocks allows for highly effective learning of complex patterns, without the requirement of a rapidly growing number of parameters. The architecture's strength and flexibility have resulted in its widespread use in both academic research and real-world applications, ranging from simple image classification tasks to more intricate scenarios such as object detection and segmentation.

ResNet50 is a pre-trained model that is highly regarded for its innovative architecture and proven track record in successfully tackling various image recognition tasks. With its pre-trained weights and adaptable architectural setup, it offers a robust tool for investigating new machine learning problems and implementing solutions in environments that prioritize accuracy and processing efficiency.

2.6 Fast Gradient Sign Method (FGSM)

The Fast Gradient Sign Method (FGSM) is a widely recognized technique used for adversarial attacks. It creates adversarial examples by introducing perturbations that are determined by the sign of the gradient of the loss in relation to the input image, multiplied by a small scalar ε (Muncsan & Kiss, 2021). This approach utilizes the gradients of the neural network to generate an image that maximizes the loss of a classifier. In simple terms, this attack functions by manipulating the original input image in a way that would amplify the prediction error.

FGSM is a cost-effective and easy-to-implement method, making it highly attractive for assessing the resilience of neural networks. By tweaking the value of ε, you can fine-tune the perturbation to evaluate different levels of robustness. When ε is small, the changes to the image are usually less noticeable, whereas larger values of ε can lead to misclassification but may also make the perturbations more noticeable to human observers.

Benefits of FGSM for Transfer Learning

When it comes to transfer learning, where a model trained on one task is repurposed for another related task, FGSM can be incredibly valuable in improving model robustness. By exposing the model to a wider range of input scenarios, there is a possibility of enhancing its ability to generalize when applied to different tasks. Transfer learning typically requires adjusting a pre-trained model on a fresh dataset or task. By incorporating adversarial training techniques like FGSM, we can prevent the model from becoming too focused on the idiosyncrasies of the new training data. This ensures that the model can maintain its performance across a wide range of data distributions, even those that it has not encountered before. In addition, developers can enhance the security of AI systems by identifying and addressing potential vulnerabilities, especially in applications where the reliability of the model is of utmost importance (Muncsan & Kiss, 2021).

The combination of being resistant to adversarial examples and improving generalization makes FGSM a valuable technique for developing and refining transfer learning models.

2.7 CIFAR10

The CIFAR-10 dataset is widely used in the field of machine learning and computer vision research. The dataset consists of 60,000 color images, each measuring 32x32 pixels. These images are evenly distributed across 10 different classes, with 6,000 images in each class. The dataset is divided into 50,000 training images and 10,000 test images, allowing for the training of machine learning models and the assessment of their performance (Krizhevsky, 2012).

Benefits of FGSM for Transfer Learning

1. **Variety and Complexity**: With a diverse range of objects like airplanes, birds, ships, and trucks, CIFAR-10 poses classification challenges due to their unique and sometimes similar features. This flexibility allows for the creation and evaluation of models that can apply to various visual domains.
2. **Standard Benchmark**: Because of its extensive history and widespread adoption in the machine learning field, CIFAR-10 has become a widely accepted benchmark for researchers to evaluate and compare the effectiveness of various algorithms in a standardized environment. This standardization contributes to the advancement of more sophisticated and precise image classification techniques.
3. **Manageable Size**: With a compact image size of 32x32 and a manageable number of images, researchers and practitioners can easily train models, making it perfect for rapid prototyping and testing. This accelerated training process allows for more frequent testing and refinement cycles in comparison to larger datasets such as ImageNet.
4. **Preprocessing and Augmentation**: The consistent size of the CIFAR-10 images makes preprocessing easier and enables the straightforward application of different data augmentation techniques. Applying data augmentation techniques can greatly improve the resilience and adaptability of models trained on this dataset (Shorten & Khoshgoftaar, 2019).
5. **Educational Tool**: The simplicity and challenge of CIFAR-10 make it an ideal educational dataset for students and newcomers in the field of machine learning and

computer vision. It offers a practical, hands-on approach to understanding neural networks, feature extraction, and the nuances of tuning and enhancing models.

Ultimately, the CIFAR-10 dataset serves a dual purpose: it facilitates the creation of sophisticated image classification models and enhances our comprehension of the underlying complexities in visual recognition tasks. In academic and research settings, it remains a valuable tool for gaining insights into the behaviour of different neural networks and machine learning algorithms (Krizhevsky, 2012).

Chapter III

3 Methodology

This chapter presents the methodology that was utilized in the project. The methodology is split into two sections. Section 3.1 presents the search methodology which includes the searching method, creation of the Inclusion- and Exclusion Criteria, The Systematic Literature Review, Snowballing, and the Search Query on Scopus. Section 3.2 presents the coding environment that was built, and the implementation of the project.

3.1 Search Methodology

The search strategy for relevant literature was carefully planned from the beginning. The process started by setting clear inclusion and exclusion criteria to guarantee the selection of high-quality, relevant academic papers. The papers had to meet specific criteria to be included. They needed to be peer-reviewed and focus on adversarial machine learning. Additionally, they had to provide detailed experimental setups with reproducible results, use publicly available datasets relevant to adversarial contexts, and involve object detection data types limited to images or videos.

On the other hand, the exclusion criteria eliminated papers that were not peer-reviewed, lacked empirical data and results, did not use relevant adversarial machine learning datasets, or failed to provide comparisons with existing machine learning methods. With these criteria in mind, the search made use of academic databases like Scopus, IEEE Xplore, and Google Scholar. By utilizing specific keywords such as "adversarial machine learning", "object detection", and "AI security", the search was narrowed down to focus on the most relevant studies. This approach ensured that only the highest quality and most applicable research was reviewed, maintaining a strong level of scholarly rigor and relevance to the thesis topic.

Throughout the progression of the master's thesis project, the search strategy for sourcing relevant literature underwent refinement and optimization. At first, the strategy had a wide range of objectives, but it was later narrowed down to solely concentrate on the Scopus database, which is renowned for its extensive collection of scholarly articles. By

utilizing Scopus's advanced search query features, this focused approach enabled more efficient navigation through extensive academic resources.

The criteria set at the beginning of the project remained in place to direct the search, guaranteeing that the literature found was both relevant and of high quality. As the user became more familiar with the database and its functionalities, the search process became more precise and efficient. Strategic utilization of keywords and subject-specific filters in Scopus greatly improved the efficiency and manageability of the research process during the thesis work, resulting in highly relevant results obtained quickly.

3.1.1 Inclusion- and Exclusion Criteria

Inclusion Criteria	
1	*The paper must be peer-reviewed*
2	*The paper must focus on adversarial machine learning*
3	*The paper must include setup for the experiments and results that are reproducible and clearly presented*
4	*Experiments must use relevant, publicly available datasets for adversarial contexts*
5	*The datatype used for object detection must be either images or videos*
Exclusion Criteria	
1	*The paper is not peer-reviewed*
2	*The study lacks experiments and results*
3	*No relevant adversarial machine learning data sets are used*
4	*There is no comparison with existing machine learning methods*

Table 1: Inclusion and Exclusion Criteria

The inclusion and exclusion criteria for the literature review are outlined in detail to ensure the selection of relevant and academically rigorous papers I shown in Table 1. All studies being considered must meet the primary inclusion criterion of being peer-reviewed. This means that only papers from recognized academic databases will be selected, ensuring a high standard of research. In addition, it is important for each paper to focus on adversarial machine learning and provide detailed experiments that can be reproduced and presented clearly. These experiments should include object detection mechanisms, using either images or videos as data types, and should make use of relevant, publicly available datasets specifically designed for adversarial contexts.

On the other hand, the criteria for exclusion are just as strict to ensure the literature review's quality. All papers must undergo a peer-review process to maintain academic credibility. Excluding papers that lack empirical data and results or fail to utilize relevant

adversarial machine learning datasets. In addition, studies that do not compare their findings with existing machine learning methods are excluded to ensure a thorough understanding of the field in the current scientific landscape. These criteria have been carefully crafted to enhance the literature review, with a specific emphasis on selecting top-notch studies that directly contribute to our understanding of adversarial tactics and defenses in machine learning for object detection.

3.1.2 Literature Review

During the development of my thesis, incorporating structured methodologies for literature reviews was essential in establishing a strong foundation for the research within the existing body of knowledge. Following the guidelines provided in "Guidance for Conducting Systematic Scoping Reviews" by Peters et al. (2015), we implemented a thorough scoping review approach. This process played a crucial role in clarifying the conceptual boundaries and identifying the central themes and gaps within the existing research on adversarial attacks in object detection (Peters et al., 2015). Using the scoping review method provided a comprehensive perspective, guaranteeing that the thesis was not only well-researched but also covered aspects that previous studies may have missed.

Building upon the thesis's foundation, valuable insights from "Systematic Literature Reviews in Software Engineering – A Tertiary Study" by Kitchenham et al. (2010) played a crucial role in improving the search strategies employed to collect relevant studies. The systematic approach recommended by Kitchenham et al. highlights the significance of conducting a thorough search process to ensure inclusiveness and reduce bias in source selection (Kitchenham et al., 2010). This approach greatly strengthened the thesis by ensuring a comprehensive and rigorous literature review. It covered a wide range of studies on adversarial machine learning and its impact on object detection.

The structured review methods had a direct impact on the quality and scope of the thesis. By following the strict guidelines outlined in these important papers, the thesis not only brought together a wide range of relevant literature but also offered fresh perspectives on the ability of DL models to withstand adversarial attacks. The methodological rigor derived from these sources was crucial in developing a thesis that had strong academic foundations and made innovative contributions to the field of adversarial machine learning

(Peters et al., 2015; Kitchenham et al., 2010). By combining systematic and scoping review techniques, the research was able to establish a strong evidence base and expand the boundaries of current knowledge.

3.1.3 Snowball

Wohlin suggests a method that involves both backward and forward snowballing. In backward snowballing, researchers review the references listed in selected papers, while in forward snowballing, they examine papers that cite the initial papers. This methodology is highly regarded for its ability to uncover literature that may be overlooked by traditional database searches. This enriches the review process and ensures a comprehensive coverage of the topic (Wohlin, 2014).

In addition, the paper discusses a replication study that utilizes these snowballing guidelines to assess their efficacy in comparison to searches driven by databases. The results confirm the effectiveness of snowballing as a strong alternative, indicating that it may even outperform traditional methods in identifying relevant studies. This supports its widespread use as a standard practice in systematic literature reviews in software engineering.

Ultimately, Wohlin's guide emphasizes the significance of systematic approaches in literature reviews and highlights the value of snowballing as a crucial tool for researchers seeking to conduct comprehensive and extensive reviews. This approach ensures a more comprehensive understanding and synthesis of existing research in a specific field.

3.1.4 QUERY on Scopus

Scopus[1] is a comprehensive and versatile research database that is highly regarded for its ability to explore scholarly literature. It is particularly valuable for conducting systematic searches across a wide range of academic fields. Scopus is an invaluable resource for academic research, providing a wide range of peer-reviewed journal articles, conference papers, book chapters, and other scientific documentation from various fields of study.

For this study, Scopus proved to be an invaluable resource, allowing for a comprehensive literature review thanks to its advanced search capabilities. These

[1] Scopus: http://www.scopus.com

capabilities are essential for researchers who want to gain a comprehensive understanding of the complex world of adversarial attacks in object detection. The database allows for the creation of intricate search queries that can be fine-tuned to cater to specific research requirements. This feature played a crucial role in filtering through a large amount of information to meet the rigorous criteria set for this thorough review.

Search Query

The search query function in Scopus enables users to specify and fine-tune their searches using a range of parameters, such as keywords, author names, publication titles, and dates. This function is improved by using Boolean operators to narrow down searches to the most relevant articles. Throughout this study, the search method was consistently improved, utilizing Scopus's powerful search capabilities to enhance the quality and relevance of the literature reviewed. Through careful adherence to a specific set of criteria, the search within Scopus was customized to produce a focused collection of work that directly aligned with the study's goals.

Approaching the search within Scopus in a methodical manner was essential in gathering a targeted selection of scholarly works. This enabled a comprehensive grasp of the existing knowledge and areas of research that need further exploration in the field of adversarial attacks and their effects on DL models. By fine-tuning and enhancing the search query on Scopus, the literature review was kept precise and current, making a direct and valuable contribution to the thorough analysis of the selected research field.

The search query was utilized to gather relevant papers for the thesis:

TITLE ("adversarial attack" OR "adversarial example*" OR "adversarial patch*" OR "adversarial defence" OR "adversarial robustness" OR "adversarial learning") ABS ("tensorflow" AND "machine learning" OR "dataset" OR "data set" OR "object detect*") AND PUBYEAR > 2016 AND PUBYEAR < 2024 AND (EXCLUDE (DOCTYPE , "cr")) AND (LIMIT-TO (LANGUAGE , "English"))*

This search query seen above is designed to retrieve scholarly articles from the Scopus database that focus on various aspects of adversarial techniques in machine learning, particularly in the context of object detection, using TensorFlow or relevant datasets. The query specifically targets articles with titles containing phrases like "adversarial attack," "adversarial example," "adversarial patch," "adversarial defence," "adversarial robustness,"

or "adversarial learning." It further refines the search to include articles where the abstract mentions "TensorFlow" and "machine learning," or any form of "dataset" or "data set," or phrases that start with "object detect" (like "object detection" or "object detecting"). Additionally, the search is limited to articles published between the years 2017 and 2023, inclusive, ensuring the results are relevant and recent. The query excludes documents classified as conference reviews ("cr") and limits the results to those published in English. This specific configuration of search terms and filters aims to capture a focused set of publications that discuss the intersection of adversarial methods and object detection technologies, providing insights into recent advancements and applications in the field.

3.2 Coding Methodology

Because of their large libraries and GPU calculation power, Python and TensorFlow, versions 3.9.16 and 2.10.0, respectively, were chosen for the coding framework's computational efficiency. Because of its complexity and suitability for evaluating adversarial robustness, CIFAR10 was selected as the main dataset; MNIST and NIFS2017 datasets offered more dimensions for testing, but was not tested further. To obtain a thorough knowledge of the datasets, pre-processing methods such as normalisation and augmentation were carefully conducted, and data visualisation tools were utilised. The models were carefully chosen because of their track record of success in object detection. Each model underwent extensive training and evaluation, with hyperparameters adjusted and visualisation tools employed to examine and analyse the models' vulnerability to adversarial cases.

3.2.1 Hardware and Software Specifications

For the experiments in my thesis, I've decided to use the NVIDIA RTX 2070 Super GPU that's installed in my personal computer. The graphical processing unit is designed to take advantage of its exceptional computational speed, especially for handling extensive datasets and intricate calculations needed in deep learning models.

Using Anaconda3, I set up a virtual environment to effectively handle and control project-specific dependencies. This method enables a tidy work environment, separate from other projects or generic configurations on my computer.

These are the essential software and libraries needed to enable GPU acceleration in the computational environment:

- **CUDA Toolkit 11.8**: This suite of development tools from NVIDIA allows for the creation and execution of applications on systems with graphics processing units.
- **cuDNN Version 8.1**: The CUDA Deep Neural Network library provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers.
- **Python 3.9.16**: The programming language chosen for its extensive support and compatibility with data science and machine learning libraries.
- **TensorFlow 2.10.0**: An open-source framework that facilitates the development and training of deep learning models with robust GPU support.

One of the main advantages of using the GPU instead of a CPU is its superior capability to handle parallel computations. This capability is essential for decreasing the time needed to train intricate machine learning models and for handling the extensive computations associated with deep learning.

Through the configuration of this specialized environment, my goal is to enhance the efficiency and performance of model training sessions conducted during this research.

3.2.2 Flowchart

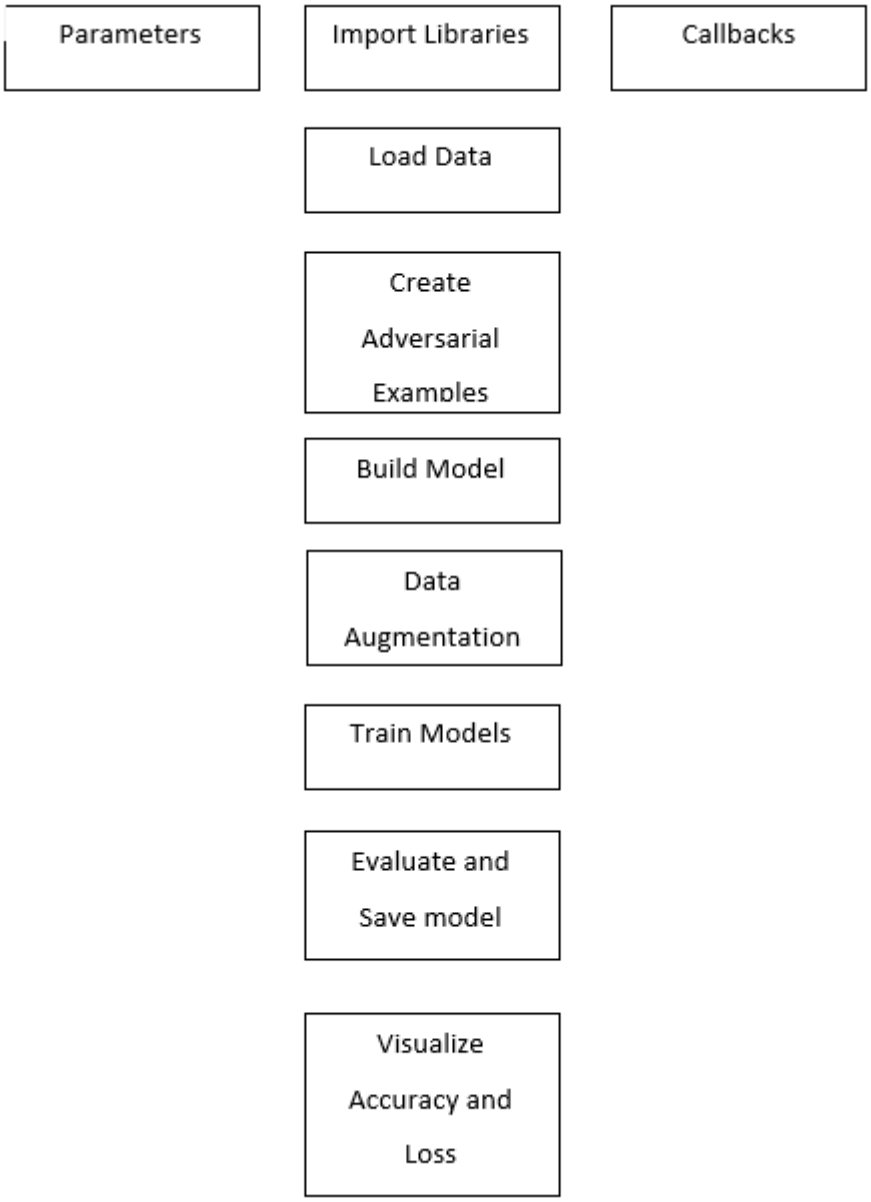

Figure 3: Flowchart of the ML model Development Process

The flowchart demonstrates the systematic approach used in the development and evaluation of machine learning models, with a focus on incorporating adversarial training techniques to improve model robustness. This process guarantees a methodical progression from the initial setup to the final evaluation, which is essential for ensuring reproducibility and conducting comprehensive experimentation.

1. **Parameters**: Establish the initial parameters that will control the model training and evaluation procedures. These factors to consider are the learning rates, batch sizes, and the number of epochs for model training.
2. **Import Libraries**: Load the essential Python libraries like TensorFlow, Keras, and NumPy. These libraries are crucial for creating and manipulating machine learning models.
3. **Callbacks**: Define callback functions such as EarlyStopping and ReduceLROnPlateau. These callbacks play a crucial role in optimizing the training process by preventing overfitting and dynamically adjusting the learning rate.
4. **Load Data**: Import the dataset that will be used for training and testing the model. The data loading step is crucial for preprocessing and getting the data ready in appropriate formats for machine learning tasks.
5. **Create Adversarial Examples**: Create adversarial examples using techniques like the Fast Gradient Sign Method (FGSM). This step is intended to improve the model's ability to generalize by learning from examples that introduce slight changes with the goal of misleading the model's predictions.
6. **Build Model**: Design the structure of the neural network. Setting up different layers, activation functions, and possibly dropout layers is necessary to create a comprehensive model that can learn from both genuine and adversarial examples.
7. **Data Augmentation**: Performing transformations on the training data enhances the variety of the training samples, thereby enhancing the model's capacity to generalize to unfamiliar data.
8. **Train Model**: Train the machine learning model using both authentic and adversarial examples. This step uses the defined callbacks and parameters set earlier to optimize the training phase.
9. **Evaluate and Save Model**: Once the training is complete, it is important to assess the model's performance by testing it on a separate dataset. This will help determine its accuracy and how well it can handle different scenarios. Ensure the trained model is saved for future use or additional analysis.
10. **Visualize Accuracy and Loss**: Visualize the training and test accuracy and loss throughout the training epochs. This visualization is useful for gaining insight into the

model's learning progression and detecting potential problems such as overfitting or underfitting.

3.2.3 Dataset

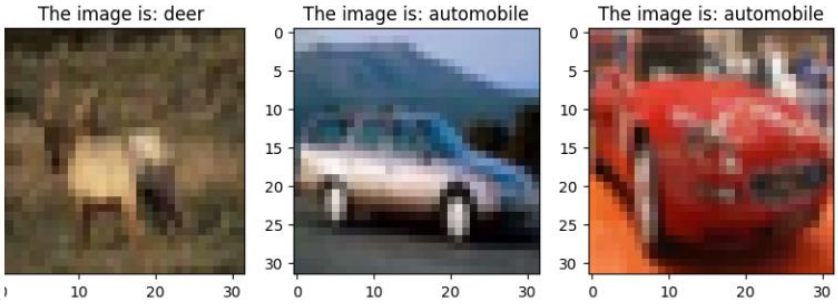

Figure 4: Visualization of Data set images.

After careful consideration, I decided to use the CIFAR-10 dataset for the thesis project. It was the most appropriate choice given the time and resource constraints of an academic setting. At first, I also thought about using the MNIST and NIST Fingerprint (NIST2017) datasets, which are highly respected in the research community for evaluating image processing and machine learning algorithms.

Decision Process

CIFAR-10: The CIFAR-10 dataset offers a well-balanced challenge with its 60,000 32x32 color images across 10 categories. It strikes a good balance between complexity and manageability. Its compact image size enables the training of advanced models without overwhelming computational demands, making it a great fit for the project's hardware constraints. In addition, the wide range of images in CIFAR-10 allows for a thorough examination of model performance across diverse visual content, resulting in valuable insights within a short project timeframe.

MNIST: Although MNIST, a dataset of 70,000 28x28 grayscale images of handwritten digits, is often chosen for its simplicity and widespread use in educational settings, it offers a less varied challenge compared to CIFAR-10. While MNIST is useful for quickly training and testing models, it only focuses on digit recognition and may not fully utilize the potential of modern image classification models.

NIST Fingerprint (NIST2017): Although MNIST, a dataset of 70,000 28x28 grayscale images of handwritten digits, is often chosen for its simplicity and widespread use in educational settings, it offers a less varied challenge compared to CIFAR-10. While MNIST is useful for quickly training and testing models, it only focuses on digit recognition and may not fully utilize the potential of modern image classification models.

Rationale behind Final Choice:

The decision to move forward with CIFAR-10 was influenced by the aim to find a middle ground between the complexity of the model and the practical limitations of computational resources and project timeline. CIFAR-10 provides a well-rounded collection of images that are ideal for conducting reliable machine learning experiments without putting too much strain on computational resources. Despite their merits, MNIST and NIST2017 were considered less suitable for the project due to the strict time and memory limitations. These limitations could potentially hinder the exploration of more complex classification tasks given the available conditions.

This strategic decision ensures that the project stays within practical limits while still providing a challenging environment to test and enhance advanced image classification techniques.

3.2.4 Loading the dataset

In the figure below, we easily load the dataset for further use.

```
# Load the dataset and print Training and Test set size
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']

print("Training set size: ", np.shape(x_train))
print("Test set size: ", np.shape(x_test))
```

executed in 399ms, finished 08:36:33 2024-08-14

```
Training set size:  (50000, 32, 32, 3)
Test set size:  (10000, 32, 32, 3)
```

Figure 5: Loading the dataset

3.2.5 Dataset Image Size

For my thesis project, I made crucial decisions about the dataset and neural network architectures to enhance the performance and feasibility of image classification tasks. One of the important factors taken into account was the size of the images in the CIFAR-10 dataset and how well different convolutional neural network (CNN) architectures can handle different image sizes.

CNN Compatibility with CIFAR-10

Conventional CNN architectures are perfectly suited to handle the 32x32 resolution of CIFAR-10 images. These models, including specialized CNNs developed for this study, are able to efficiently extract the key features from small images. This allows for faster training times and optimized use of memory resources. Nevertheless, when contemplating the incorporation of advanced architectures such as MobileNetV2 and ResNet50, the limited image size poses certain constraints on the ability to extract features.

Upscaling for MobileNetV2 and ResNet50

Given that MobileNetV2 and ResNet50 are designed to work with higher resolution inputs, typically 96x96 pixels or larger, it was necessary to upscale the CIFAR-10 images to meet these requirements. To address this, I integrated a Conv2DTranspose layer within the neural network architecture to dynamically upscale the input images from 32x32 to 96x96 pixels. This approach ensures that the images are adequately resized to leverage the full potential of MobileNetV2 and ResNet50, allowing these advanced models to effectively extract complex features and improve classification accuracy.

3.2.6 Models Overview

a. Sequential Convolutional Neural Network

```
Model: "sequential_12"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_31 (Conv2D)          (None, 32, 32, 32)        896

 batch_normalization_25 (Bat  (None, 32, 32, 32)       128
 chNormalization)

 conv2d_32 (Conv2D)          (None, 30, 30, 32)        9248

 average_pooling2d_3 (Averag  (None, 15, 15, 32)       0
 ePooling2D)

 dropout_26 (Dropout)        (None, 15, 15, 32)        0

 conv2d_33 (Conv2D)          (None, 15, 15, 64)        18496

 batch_normalization_26 (Bat  (None, 15, 15, 64)       256
 chNormalization)

 conv2d_34 (Conv2D)          (None, 13, 13, 64)        36928

 average_pooling2d_4 (Averag  (None, 6, 6, 64)         0
 ePooling2D)

 dropout_27 (Dropout)        (None, 6, 6, 64)          0

 conv2d_35 (Conv2D)          (None, 6, 6, 256)         147712

 batch_normalization_27 (Bat  (None, 6, 6, 256)        1024
 chNormalization)

 conv2d_36 (Conv2D)          (None, 4, 4, 256)         590080

 average_pooling2d_5 (Averag  (None, 2, 2, 256)        0
 ePooling2D)

 dropout_28 (Dropout)        (None, 2, 2, 256)         0

 flatten_12 (Flatten)        (None, 1024)              0

 dense_23 (Dense)            (None, 512)               524800

 dropout_29 (Dropout)        (None, 512)               0

 batch_normalization_28 (Bat  (None, 512)              2048
 chNormalization)

 dense_24 (Dense)            (None, 10)                5130

=================================================================
Total params: 1,336,746
Trainable params: 1,335,018
Non-trainable params: 1,728
_________________________________________________________________
```

Figure 6: Model Architecture for the CNN.

This model architecture is a simple convolutional neural network (CNN) specifically designed for image classification tasks. It begins with multiple convolutional layers, each followed by batch normalization to stabilize the learning process by normalizing the activations. This model incorporates three sets of Conv2D and Batch Normalization layers, which enhances the depth of the network to effectively capture intricate features. Pooling layers help to decrease the computational load for subsequent layers by reducing the spatial dimensions of the feature maps. Dropout layers are strategically placed to prevent

overfitting by randomly eliminating units during the training process. The network ends with fully connected layers that flatten the output of the convolutional bases to generate the final class predictions. This structure is highly efficient in managing diverse and intricate image data, enhancing both the extraction of features and the accuracy of classification.

b. MobileNetV2

```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenetv2_1.00_128 (Funct  (None, 4, 4, 1280)       2257984
 ional)

 global_average_pooling2d_12  (None, 1280)             0
  (GlobalAveragePooling2D)

 reshape_10 (Reshape)        (None, 1, 1, 1280)        0

 Dropout (Dropout)           (None, 1, 1, 1280)        0

 conv2d_6 (Conv2D)           (None, 1, 1, 10)          12810

 softmax (Activation)        (None, 1, 1, 10)          0

 flatten_7 (Flatten)         (None, 10)                0

=================================================================
Total params: 2,270,794
Trainable params: 2,236,682
Non-trainable params: 34,112
```

Figure 7: Model Architecture for MobileNetV2.

This model uses MobileNetV2, a lightweight deep neural network architecture optimized for mobile and edge devices, with a focus on performance efficiency. The architecture utilizes depthwise separable convolutions to greatly decrease the parameter count while maintaining high performance, as seen in traditional CNNs. In this particular configuration, the MobileNetV2 base is accompanied by a global average pooling layer, which effectively reduces each feature map to a single value. This approach helps prevent overfitting and also decreases the computational complexity. This is combined with a dropout layer for regularization and a final dense layer with softmax activation to generate

the probabilities of the ten class labels. This configuration is perfect for situations where computational resources are scarce, but there is a need for precise image classification.

c. ResNet50

```
Model: "sequential_15"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 6, 6, 2048)        23587712

 global_average_pooling2d_15  (None, 2048)             0
  (GlobalAveragePooling2D)

 flatten_15 (Flatten)        (None, 2048)              0

 dense_27 (Dense)            (None, 512)               1049088

 dense_28 (Dense)            (None, 10)                5130

=================================================================
Total params: 24,641,930
Trainable params: 1,054,218
Non-trainable params: 23,587,712
_________________________________________________________________
```

Figure 8: Model Architecture for ResNet50.

This model is based on ResNet50, which is renowned for its deep residual learning framework that allows for the training of networks that are much deeper than previous ones. ResNet50 relies on the clever use of residual blocks that include skip connections, allowing for seamless jumps over certain layers. These connections are crucial in addressing the vanishing gradient problem by enabling the direct backpropagation of the gradient to earlier layers. The architecture concludes with a global average pooling layer and a dense layer setup, which is commonly used for classification tasks. Using a highly complex network like ResNet50 is ideal when achieving utmost accuracy is crucial and the computational resources required are justified by the enhanced ability to extract more detailed features.

3.2.7 Optimizer Adam

The Adam optimizer is highly regarded in the field of deep learning due to its ability to adaptively adjust learning rates, effectively converge, and remain stable even when hyperparameters are modified. This makes it particularly well-suited for handling intricate models and extensive datasets. It combines the benefits of AdaGrad and RMSProp

optimizers by adapting learning rates using gradient mean and uncentered variance, improving performance in scenarios with sparse or noisy gradients. Adam also incorporates an automatic bias correction feature, ensuring optimal performance from the beginning and throughout the training process. This is demonstrated in the model compilation line:

model_2.compile(optimizer='adam',

loss='categorical_crossentropy',metrics=['accuracy']).

Adam is used to optimize a multi-class classification model, resulting in high accuracy and efficient training in various learning scenarios.

3.2.8 Categorical_crossentropy

The categorical_crossentropy loss function is commonly employed in multi-class classification problems where each target class is represented using one-hot encoding. This function calculates the performance of a classification model that produces a probability value ranging from 0 to 1. The categorical_crossentropy function calculates the loss by comparing the predicted probabilities with the true distribution. It essentially measures the negative logarithm of the probability assigned to the correct class. This metric promotes the model to prioritize assigning higher probabilities to the correct class labels, resulting in more accurate predictions. Through penalizing the divergence of probabilities from the actual class labels, this approach guarantees that the model's predictions will converge to the true distribution. This makes it a great option for tasks that involve distinguishing between multiple categories.

3.2.9 Metrics

In our instance where classes are to be predicted, we utilized the following metrics:

Accuracy

As the percentage of accurately predicted occurrences among all instances, accuracy is a crucial parameter in classification tasks. It is determined by dividing the total number of forecasts by the number of accurate guesses. Accuracy is a simple and obvious metric, but in situations where there is a class imbalance—that is, when the majority class dominates the predictions and the model does badly on the minority class—it can be deceptive.

Loss

A model's loss is a measurement of how closely or not its predictions during training or testing match the real labels. By comparing the predicted probability to the true labels, loss functions such as categorical cross-entropy are frequently employed in supervised learning to quantify the inaccuracy. Reducing the loss is the aim of training a model, which raises the predicted accuracy of the model. A model that produces forecasts that are more accurate is indicated by a lower loss; a model that produces predictions that are more inaccurate is indicated by a larger loss.

Precision

Precision measures how well the model predicts favourable outcomes. The ratio of true positive predictions to all positive predictions (true and false positives) is how it is defined. Since precision indicates the proportion of correctly anticipated positive cases, it is especially crucial in situations when the cost of false positives is large. A high precision means that there aren't many false positives in the model.

AUC

Area Under the Receiver Operating Characteristic (ROC) Curve, or AUC, is a single statistic that expresses how well the model can differentiate across classes. An AUC number that is closer to 1 suggests a model that has a high degree of capacity to discriminate between positive and negative classes. The AUC goes from 0 to 1. An AUC of 0.5 is comparable to random guessing and indicates no discriminative ability. When assessing models on unbalanced datasets, the AUC is very helpful since it takes into account the trade-off between the True Positive Rate (Recall) and the False Positive Rate.

Recall

Recall, often referred to as True Positive Rate or Sensitivity, gauges a model's capacity to locate every pertinent instance in a dataset. It is computed as the ratio of the total number of real positives (true positives and false negatives) to the true positive forecasts. Recall is important when it comes to situations when it is more important to miss a positive instance (false negative) than it is to incorrectly guess a negative occurrence to be

positive. When a recall value is high, it means that the majority of positive cases are correctly captured by the model.

F-measure (F1 Score)

The F-measure, or F1 Score, is the harmonic mean of precision and recall, providing a single metric that balances both. It is particularly useful when the dataset is imbalanced, as it ensures that both false positives and false negatives are considered. The F1 Score ranges from 0 to 1, with a value closer to 1 indicating a better balance between precision and recall. This metric is favored when we seek a balance between avoiding false positives and false negatives, rather than optimizing for either one alone.

Confusion Matrix

A Confusion Matrix is a table that summarizes the performance of a classification model by displaying the actual versus predicted classifications. It comprises four key components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The confusion matrix offers detailed insight into the model's performance, highlighting not just overall accuracy but also the types of errors being made. It is particularly useful in understanding how well the model performs on each class, especially in multiclass classification problems.

3.2.10 EarlyStopping

The EarlyStopping callback is used to prevent overfitting during the training process. It monitors a specified metric (in this case, val_loss which is the test loss) and stops the training if this metric does not improve after a certain number of epochs.

- **monitor='val_loss'**: Specifies that the validation loss is the metric to be monitored.
- **patience=10**: The model will stop training if the test loss does not improve for 10 consecutive epochs. This helps in stopping the training early if the model has stopped learning.
- **restore_best_weights=True**: Once training is stopped, the model will revert to the weights from the epoch that had the best test loss. This ensures that you have the

best model parameters even if training continued for a few more epochs after the optimal point.

3.2.11 ReduceLROnPlateau

The ReduceLROnPlateau callback is used to reduce the learning rate when the metric monitored (again, val_loss in this case) stops improving. Reducing the learning rate can help the model converge more effectively, especially when the loss plateaus.

- **monitor='val_loss'**: The callback monitors the validation loss.
- **factor=0.2**: When triggered, the learning rate is reduced by a factor of 0.2 (i.e., the learning rate is multiplied by 0.2).
- **patience=5**: If the test loss does not improve for 5 consecutive epochs, the learning rate will be reduced.
- **min_lr=1e-6**: This sets a lower bound on the learning rate. The learning rate will not be reduced below this value, ensuring it doesn't become too small.

Chapter IV

4 Results

4.1 Results from *Table 2: Literature Study Table* Literature Review

4.1.1 Snowballing

4.1.2 Adversarial Attacks

Title	Snowball	YEAR	ML Model	Attack type	Domain	Dataset	Evaluation Metrics/Goal
Generating Black-Box Adversarial Examples in Sparse Domain(Zanddizari et al., 2022)	No	2022	Deep Learning, Efficient-B1	Black-box Adversarial Attacks, *Untargeted*	Image recognition	Efficient-B1 and MNIST	Accuracy and Misclassification rates
Towards Selective Adversarial Attack for Gait Recognition Systems based on DNN (Kwon, 2023)	No	2023	Deep Learning, CNN with ReLU	Evasion Attacks, Targeted	Gait Recognition (Image recognition)	CASIA Gait Database B	Accuracy and Attack Success Rate
Adverarial Attacks on Visual Objects using FGSM (Naqvi et al., 2023)	No	2023	Deep Learning, MobileNetV2	Gradient-based attacks	Image Recognition	ImageNet	Misclassification rate
Priority Evasion Attack: An Adversarial Example that considers the Priority of Attack on Each Classifier (Kwon et al., 2019)	Yes	2022	Deep Learning	Evasion Attacks, Targeted and Untargeted	Image Recognition	CIFAR10 and MNIST	Accuracy
Fooling a NN in Military Environments: Random Untargeted Adversarial Example (Kwon et al., 2018)	Yes, From Priority Evasion Attack	2018	Deep Learning, CNN	Evasion Attacks, Untargeted	Image Recognition	CIFAR10 and MNIST	Generates images
Diversity Adversarial Training Against Adversarial Attack on DNN (Kwon & Lee, 2021)	No, but same authors as two previous	2021	Deep Learning, CNN	Evasion Attacks, Targeted and Untargeted	Image Recognition	MNIST, Fashion-MNIST	Accuracy
Restricted-Area Adversarial Example Attack for Image Captioning Model (Kwon & Kim, 2022)	No, but same authors as two previous	2022	Deep Learning, CNN and RNN	Evasion Attacks, Targeted and untargeted, White-box	Image Recognition, Image Captioning model	MS COCO, FLICKR 8K	Attack success rate, classification results, Untargeted attacks
Adversarial Explanations for Understanding Image Classifications Decisions and Improved Neural Network Robustness (Woods et al., 2019)	No	2019	Deep Learning, CNN for Image Classification, RNN for Sequential Data Processing	Evasion Attacks	Image Recognition, Classifying and Understanding image content		

4.1.3 Table Description – Snowballing and Adversarial Attacks

Table 2 – Snowballing

Title	YEAR	Method	Attack type	Domain	Detector
Adversarial Examples for Semantic Segmentation and Object Detection (Xie et al., 2017)	2017	It introduces a Dense Adversary Generation (DAG) algorithm to create adversarial examples targeting both semantic segmentation and object detection.	The paper's approach aligns with a white-box attack framework, as it assumes access to the model's structure during the adversarial example generation.	The study focuses on digital domain attacks, manipulating images to test against deep learning models in computational environments.	The targets are deep learning models used for semantic segmentation and object detection, likely including both one-stage and two-stage detectors.
Fooling the Eyes of Autonomous Vehicles: Robust Physical Adversarial Examples Against Traffic Sign Recognition Systems(Jia et al., 2022)	2022	The paper introduces a systematic approach to create physical adversarial examples (AEs) to deceive real-world object detectors. It includes methods like image transformation and bounding box filters to generate robust adversarial perturbations.	This research conducts black-box attacks, as it assumes the attacker does not have access to the internal details of the target TSR system. It includes four types of attacks: Hiding Attack, Appearance Attack, Non-Target Attack, and Target Attack.	The attacks are conducted in the physical domain, specifically targeting real-world scenarios like fooling autonomous vehicles' TSR systems with manipulated traffic signs.	It targets object detectors used in Traffic Sign Recognition (TSR) systems, particularly focusing on YOLO v5, which is a one-stage detector.
An Enhanced Transferable Adversarial Attack Against Object Detection(Shi et al., 2023)	2023	The paper focuses on enhancing the transferability of adversarial examples in object detection. It uses a novel method that attacks both the feature map of the backbone and the detection head of CNN-based object detectors.	The approach is primarily a white-box attack, where the authors have access to the model's architecture and parameters. However, the enhanced transferability implies effectiveness in black-box scenarios as well.	The attacks are digital, targeting object detection models in a computational setting, specifically designed for adversarial example generation.	The study involves attacking CNN-based object detection systems, including both one-stage and two-stage detectors.
Object Tracking and Detection Techniques under GANN Threats: A Systemic Review (Al Jaberi et al., 2023)	2023	It discusses generative adversarial neural networks (GANNs) and their applications in object detection and tracking, focusing on their use in adversarial attacks.	Various types of attacks are reviewed, including white-box, black-box, and grey-box attacks, each with different levels of knowledge about the system being attacked.	The study encompasses both digital and physical domain attacks, considering the impact of GANN threats in real-world surveillance and object tracking applications.	The paper reviews various object detection and tracking techniques, likely including both one-stage and two-stage detectors, as well as other machine learning models.

Table 3: Adversarial Attacks

This table presents a range of studies that examine adversarial examples in various domains, including image recognition and gait recognition. The studies explore different deep learning models, attack types, and evaluation metrics. Every entry provides information on the title, publication year, machine learning model utilized, attack type (such as black-box or evasion), specific domain, dataset used, and evaluation metrics or goals. This table provides a comprehensive overview of the evolution and changes in adversarial attack research, showcasing various techniques that improve the resilience of models against these attacks.

Table 3 – Adversarial Attacks

This table provides a detailed analysis of various methodologies that have been developed for creating adversarial attacks. This collection features influential publications that introduce novel adversarial attack techniques or improve existing ones. These papers provide detailed insights into the methods, types of attacks, domains, and detection mechanisms employed. I have conducted extensive research on creating strong adversarial examples for semantic segmentation, object detection, and other specialized applications, such as deceiving autonomous vehicle systems.

Purpose and Utility of These Tables

These tables provide a comprehensive overview of the evolution and validation of various approaches in adversarial machine learning, offering valuable insights into the scope and depth of existing research across different applications. They play a crucial role in identifying gaps in the current knowledge base, guiding future research directions, and establishing a framework for systematic comparison of different techniques. Through summarizing this information, the tables enable researchers to efficiently compare study outcomes, methodologies, and the effectiveness of various adversarial techniques. This facilitates a well-informed discussion on the development of more secure AI systems to counter adversarial threats.

4.2 Results from Model Training and Evaluation

4.2.1 CNN Model Results and Visualization

In the figure below is the training history with the data from the CIFAR10 data set. Figure 8 shows the CNN model's accuracy, AUC and loss. The CNN model was trained in the whole 100 epochs and took 18 minutes and 8 seconds.

Accuracy

The plot shows that both training and test accuracy steadily increase over the epochs, while the test accuracy plateaus slightly below it, indicating good generalization with some slight overfitting as the model becomes increasingly accurate on the training data.

AUC

The Area Under the Curve (AUC) for both the training and test sets shows a rapid increase early in the training, reaching high values quickly. The AUC for both training and test is very close, indicating that the model is performing well in distinguishing between the classes on both sets, with minimal overfitting.

Loss

The loss for both training and test decreases sharply in the initial epochs, which is typical as the model learns to minimize errors. The training loss continues to decrease consistently, while the test loss stabilizes with some fluctuations, which may indicate a bit of overfitting as the model starts to memorize the training data.

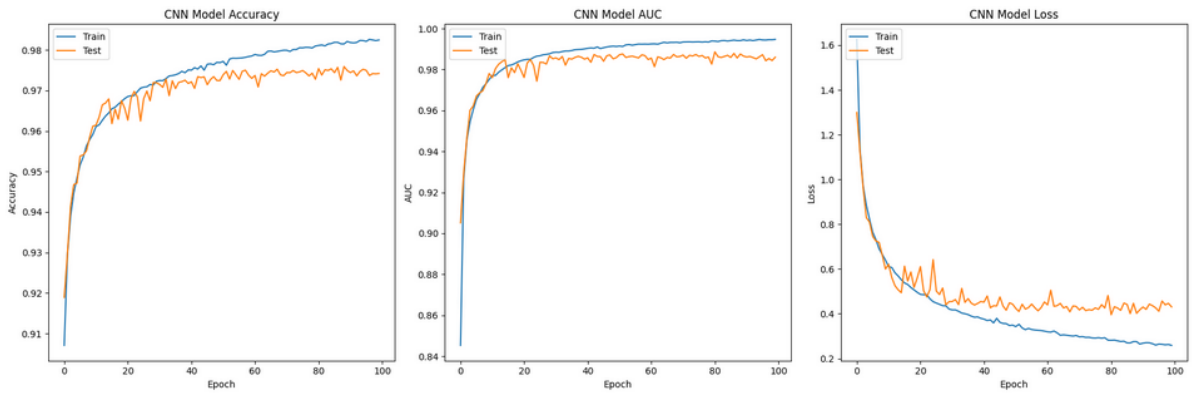

Figure 9: Plotting the history of Accuracy, AUC and Loss for the CNN model

The figure below shows the Precision and Recall history of the CNN model.

Precision

The training precision increases steadily and reaches close to 100% as the epochs progress, indicating that the model becomes increasingly confident in its correct predictions over time. The test precision also increases initially but stabilizes at a lower level compared to the training precision, with some fluctuations. This suggests that while the model performs very well on the training data, it doesn't generalize as effectively to the test data, potentially due to slight overfitting.

Recall

The training recall shows a similar trend to precision, with a rapid increase early in the training process, reaching high values as the model continues to train. The test recall starts strong but, like precision, stabilizes at a lower value than the training recall. The relatively stable yet lower recall on the test set indicates that the model is consistently identifying most of the positive instances, but again, it may not be as effective on unseen data compared to the training data.

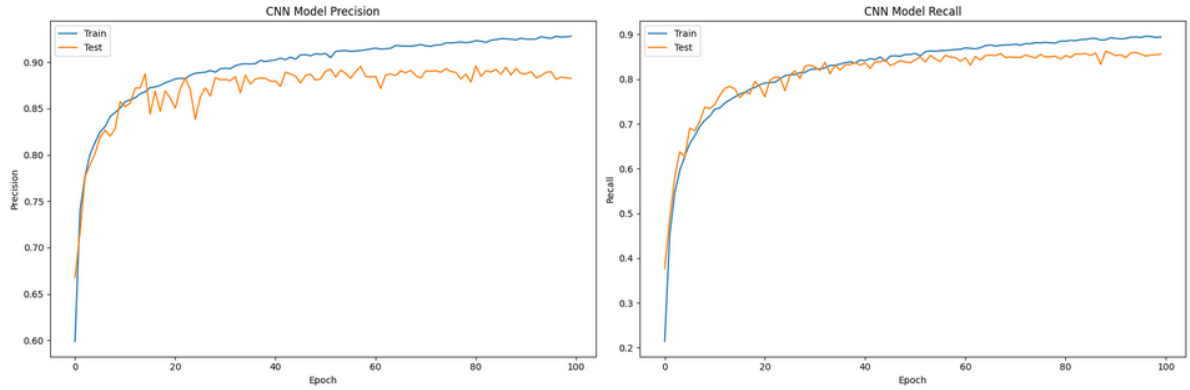

Figure 10: Plotting the history of Precision and Recall for the CNN model

The figure below is the confusion matrix for the CNN model.

Confusion Matrix

The high values along this diagonal indicate that the model is generally performing well across most classes, with particularly strong performance in classifying classes 1, 7, and 9, which have the highest correct classifications (940, 894, and 926, respectively).

Off-diagonal elements reflect misclassifications, where the true class does not match the predicted class. For instance, class 2 has 33 instances misclassified as class 0 and 48 instances misclassified as class 4. Similarly, class 3 has 37 instances incorrectly predicted as

class 6. These misclassifications suggest that the model struggles to distinguish between certain classes, particularly those with similar features.

Class 4 Misclassifications: Class 4 seems to be frequently misclassified into other classes, with 128 instances being incorrectly predicted as class 5. This indicates a potential overlap in features between these two classes that the model is finding challenging to separate.

Class 0 and 2 Mix-ups: There are notable confusions between class 0 and class 2, with 33 instances of class 2 being misclassified as class 0. This might suggest that the visual features the model uses to distinguish these classes are not sufficiently different.

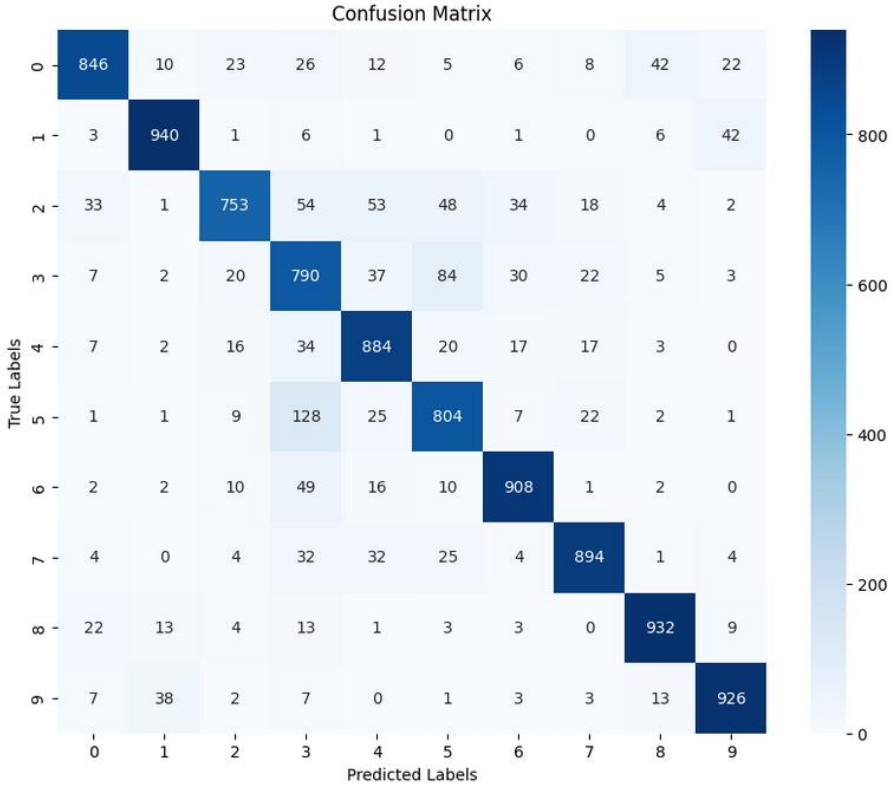

Figure 11: Plotting the Confusion Matrix for CNN model

In the figure below (Figure 12), we present the predictions on the CIFAR10 dataset from the CNN model.

Original Image Prediction: The first image depicts a correctly classified instance where the model predicted the class as 6 (which corresponds to the true class). The model's confidence in this prediction was 100%. In the second image, the model also correctly identified the original image as class 6, with a confidence level of 84%. This confidence is slightly lower than in the first instance but still indicates a strong belief in the correct classification.

Adversarial Image Prediction: An adversarial example was generated from the original image using the Fast Gradient Sign Method (FGSM). The adversarial image was still predicted as class 6 by the model, albeit with slightly reduced confidence of 92%. This indicates that the adversarial perturbation, though present, was not strong enough to fool the model in this instance. In the second image, the model's prediction was altered to class 2, which is incorrect. The confidence level for this misclassification was 100%, showing that the adversarial perturbation was highly effective in misleading the model in this case.

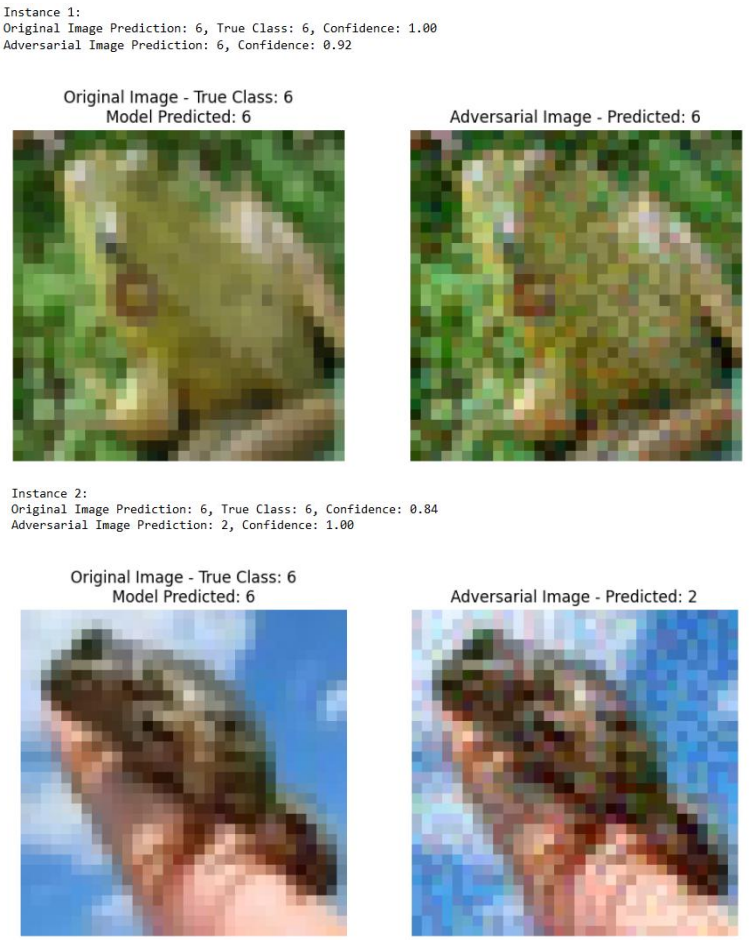

Figure 12: Instance 1 and 2 for CNN prediction

The overall results of the model's performance on the original and adversarial images are summarized as follows:

- **Accuracy on Original Images:** The model achieved an accuracy of 80% on the original test images. This indicates that 80% of the original images were correctly classified by the model without any adversarial perturbation.
- **Accuracy on Adversarial Images:** The accuracy dropped to 60% when the model was tested on adversarial images. This 20% reduction in accuracy demonstrates the model's vulnerability to adversarial attacks, as a significant portion of the images were misclassified after perturbation.
- **Average Confidence on Original Images:** The model's average confidence in its predictions for the original images was 0.93. This high confidence level suggests that the model was generally certain about its predictions when presented with unaltered data.
- **Average Confidence on Adversarial Images:** The average confidence dropped to 0.77 on the adversarial images. Although the confidence remained relatively high, the decrease reflects the uncertainty introduced by the adversarial perturbations.

```
Accuracy on Original Images: 80.00%
Accuracy on Adversarial Images: 60.00%
Average Confidence on Original Images: 0.93
Average Confidence on Adversarial Images: 0.77
```

Figure 13: Overall results of the prediction of CNN

4.2.2 MobileNetV2 Model Visualization

In Figure 11 below is the training history of the MobileNetV2 model's accuracy, AUC and loss. The CNN model was trained in only 36 epochs and took 29 minutes and 32 seconds.

Accuracy

The training accuracy rapidly increases and approaches near 100% over the epochs, indicating that the model is learning the training data very effectively. The test accuracy improves initially but then fluctuates and stabilizes at a lower level compared to the training

accuracy. This gap suggests that the model is likely overfitting, where it performs exceptionally well on the training data but not as well on the unseen test data.

AUC

The Area Under the Curve (AUC) for the training set quickly reaches close to 1, indicating excellent performance in distinguishing between classes in the training data. The test AUC also starts high and stabilizes, showing good, but not perfect, generalization. The fact that the test AUC is lower than the training AUC again points to possible overfitting, as the model might be overly tailored to the training set.

Loss

The training loss decreases steadily and approaches near zero, which is expected as the model minimizes the errors on the training set. However, the test loss starts high and decreases initially but then exhibits fluctuations and stabilizes at a higher value than the training loss. This behaviour further indicates overfitting, where the model continues to improve on the training data while its performance on the test data plateaus or worsens.

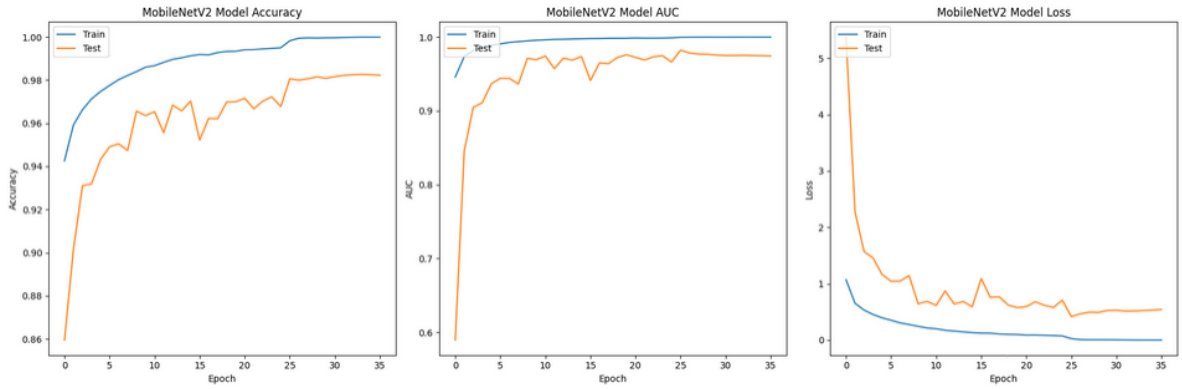

Figure 14: Plotting the history of Accuracy, AUC and Loss for the MobileNetV2 model

Precision

The precision for the training set increases steadily and approaches nearly 100% as the epochs progress, indicating that the model is increasingly confident and correct in its positive predictions on the training data. The test precision also improves over the epochs but stabilizes at a lower value compared to the training precision, with some fluctuations. This suggests that while the model performs well on the training data, it is not as confident in its positive predictions on the test set, which could be a sign of overfitting.

Recall

The recall for the training set similarly increases and reaches high levels, indicating that the model effectively identifies most of the positive instances in the training data. However, the test recall, while improving initially, also stabilizes at a lower level compared to the training recall, with some noticeable fluctuations. This suggests that the model can identify positive instances in the test set but not as effectively as it does with the training data, further pointing to potential overfitting.

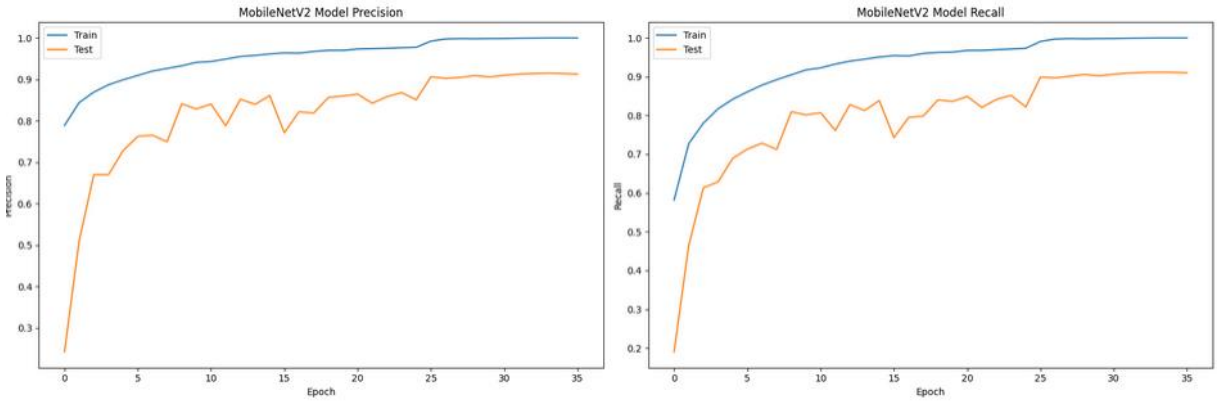

Figure 15: Plotting the Precision and Recall metrics for the MobileNetV2 Model

Confusion Matrix

In the confusion matrix, the highest values are on the diagonal, which indicates that your model is correctly classifying a large number of instances for each class.

Here, class 3 (true label) was misclassified as class 5 for 113 instances, and class 9 was misclassified as class 1 for 45 instances.

In this instance, class 1 and class 8, show particularly high diagonal values (976 and 945, respectively), suggesting strong model performance in identifying these classes. But in other areas, other classes like class 3 and class 9 have lower diagonal values.

Similar to the CNN model, the confusion matrix shows that the model is generally performing well, with most instances being correctly classified. However, there are notable misclassifications which suggests that the model could benefit from further fine-tuning to improve its ability to distinguish between these similar classes.

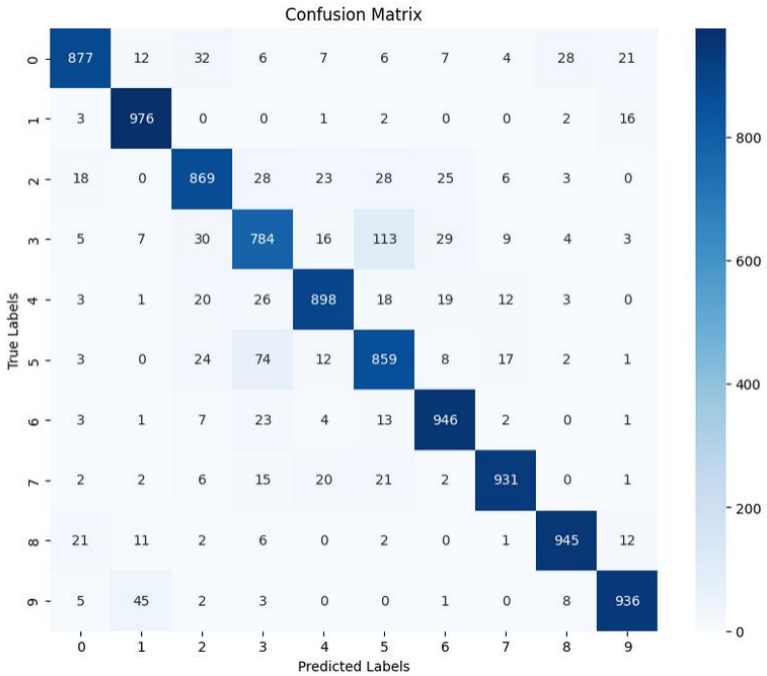

Figure 16: Plotting the Confusion Matrix for MobileNetV2

As seen in figure 17, the images and results demonstrate the model's performance on both original and adversarial perturbed images. In the first instance, the model correctly identified the original image with high confidence, but the adversarial image caused a misclassification, with the confidence dropping to 0.51. Similarly, in the second instance, while the model correctly classified the original image as class 7 with high confidence, the adversarial perturbation led to an incorrect prediction, with a significant drop in confidence.

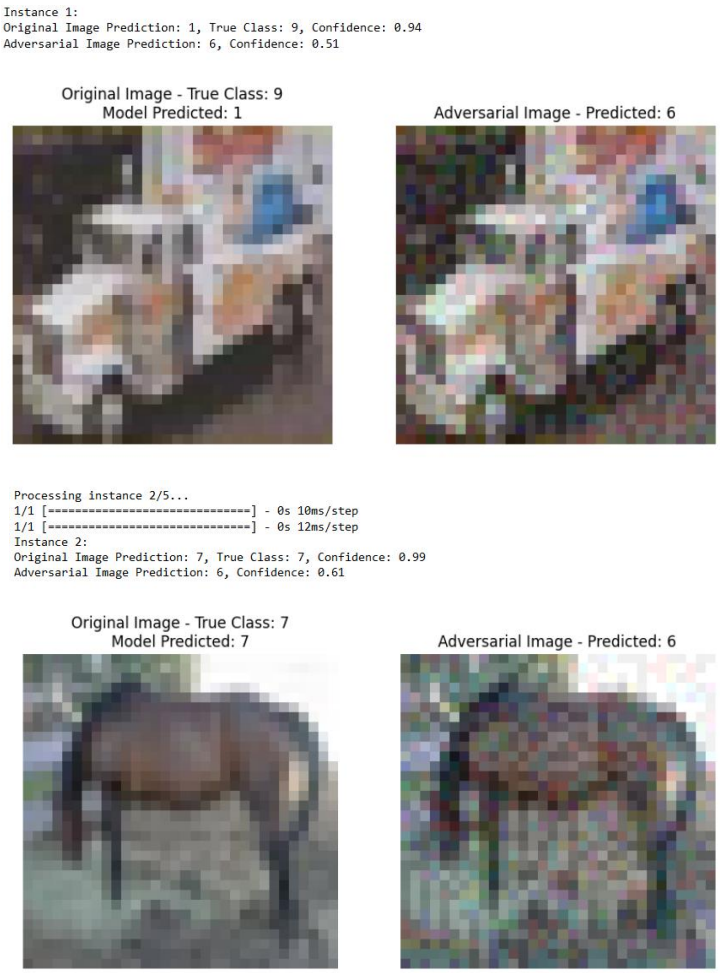

Figure 17: Instance 1 and 2 for CNN prediction

In figure 18, the model maintained 80% accuracy on the original images, showcasing its effectiveness in normal conditions. However, the model's performance dramatically deteriorated when exposed to adversarial images, with an accuracy of 0% across the tested instances. This stark contrast highlights the model's susceptibility to adversarial attacks, despite maintaining relatively high confidence levels even when making incorrect predictions.

```
Accuracy on Original Images: 80.00%
Accuracy on Adversarial Images: 0.00%
Average Confidence on Original Images: 0.96
Average Confidence on Adversarial Images: 0.76
```

Figure 18: Overall results on the predictions from MobileNetV2

4.2.3 ResNet50 Model Visualization

In the figure below is the training history for the ResNet50 model. The model was only trained for 20 epochs and took 32 minutes and 43 seconds.

Accuracy

The training accuracy increases sharply and reaches near 100% within the first 10 epochs, indicating that the model quickly learns to classify the training data correctly. However, the test accuracy plateaus much earlier and at a significantly lower value, suggesting that the model may be overfitting, where it performs very well on the training data but does not generalize as well to unseen data.

AUC

The AUC for the training set rapidly increases and stabilizes close to 1, which is typical of a model that is highly effective at distinguishing between classes in the training data. However, the test AUC, while initially rising quickly, fluctuates and then stabilizes at a lower value compared to the training AUC, which could be another sign of overfitting.

Loss

The training loss decreases steadily, approaching zero, which is expected as the model continues to minimize the errors on the training data. On the other hand, the test loss initially decreases but then fluctuates and begins to increase after a few epochs, further indicating overfitting as the model begins to perform worse on the test data after a certain point.

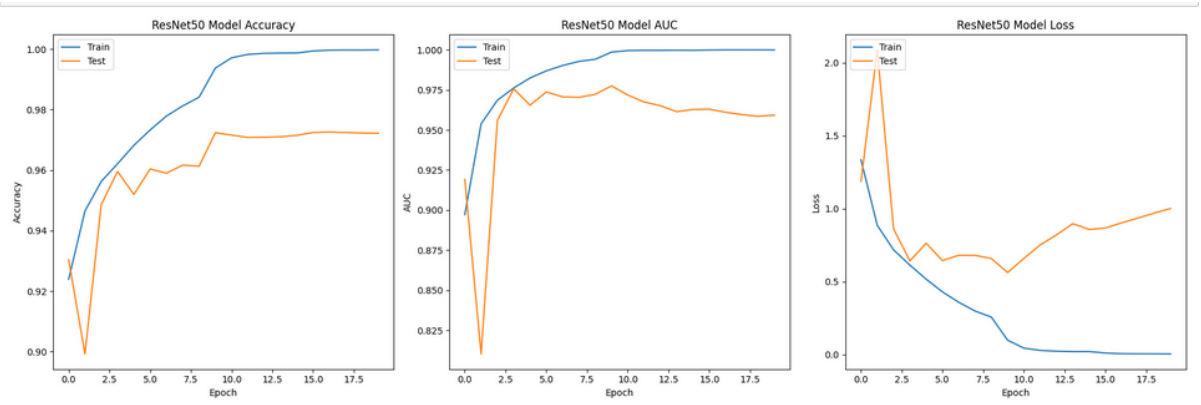

Figure 19*: Plotting the history of Accuracy, AUC and Loss for the ResNet50 Model*

In the figure below is the training history of the metrics precision and recall for the ResNet50 Model.

Precision

The precision for the training set increases sharply and continues to improve, approaching near-perfect precision as the epochs progress. This indicates that the model becomes increasingly confident in its correct predictions over time. However, the test precision stabilizes at a significantly lower value, with some fluctuations throughout the training process. This suggests that while the model performs exceptionally well on the training data, it struggles to maintain the same level of precision on the test set, which may indicate overfitting.

Recall

The training recall also improves rapidly and reaches very high levels, similar to precision, indicating that the model effectively identifies most positive instances in the training data. On the other hand, the test recall increases initially but levels off at a lower value, much like the precision. The lower recall on the test set, combined with the higher training recall, further suggests that the model may not be generalizing well to unseen data, which is often a sign of overfitting.

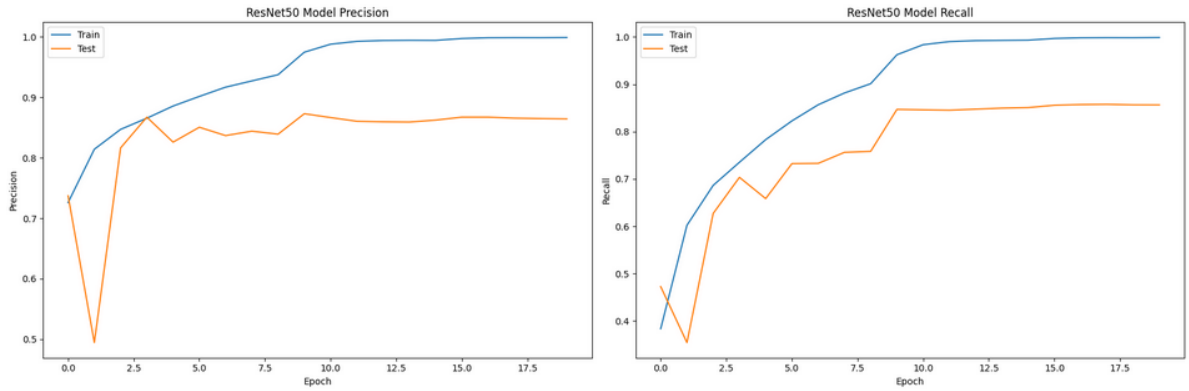

Figure 20: Plotting the Precision and Recall metrics for the ResNet50 Model

In Figure 16 below, is the confusion matrix for the ResNet50 model.

Confusion Matrix

Strong Performance: In this case, classes 1, 6, 7, and 9 show particularly strong performance, with 930, 866, 886, and 935 correct classifications respectively. This suggests that the model is highly accurate when it comes to these classes.

Inter-class Confusions: While the model performs well, certain classes are frequently confused with others. For example, class 3 has 103 instances misclassified as class 5, and class 5 itself has 125 instances misclassified as class 3. This highlights a challenge the model faces in distinguishing between these two classes, possibly due to similarities in their features.

Misclassification Trends: Class 2 is misclassified as class 0 in 26 instances and as class 4 in 35 instances. This indicates a trend where class 2 might share features with these other classes that lead to misclassification.

Focused Refinement: The matrix suggests areas where the model could be refined to improve its accuracy. For example, additional training data or feature engineering might help the model better differentiate between classes 3 and 5, which seem to be a consistent source of confusion.

Further Evaluation: Examining why certain classes are misclassified more frequently could provide insights into the model's decision-making process. It may involve exploring whether these classes have overlapping features or if there's an imbalance in the training data that affects the model's performance. The matrix reveals specific confusions, such as class 5 being often misclassified as class 3 and vice versa. These errors might be due to similarities in features between these classes that the model struggles to differentiate.

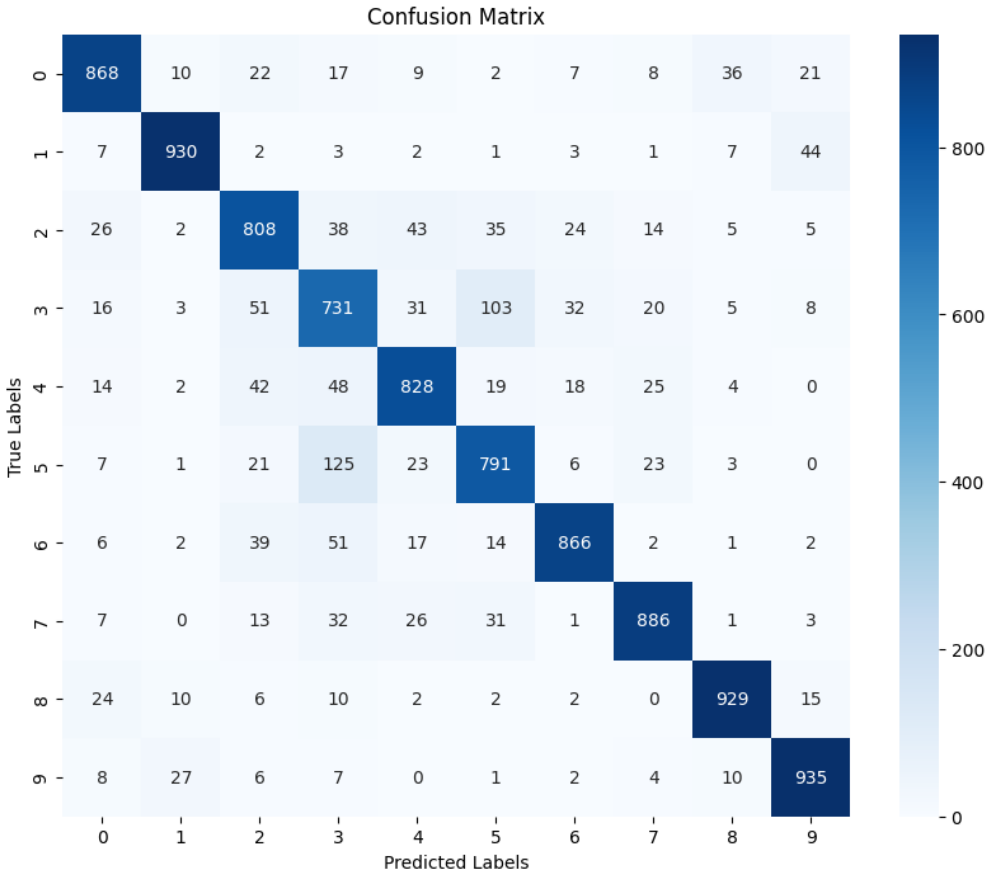

Figure 21: Plotting the Confusion Matrix for ResNet50 model

In figure 22, the model's ability to correctly classify the original images was compromised by adversarial perturbations. In the first instance, the model incorrectly identified the original image of a horse (class 7) as a dog (class 3) with low confidence, which was further reduced when exposed to the adversarial image. The second instance shows a successful classification of the original image (class 3) with high confidence, but the adversarial example misled the model into predicting a different class (class 2) with maximum confidence

Figure 22: Instance 1 and 2 from the prediction of ResNet50

As seen in figure 23, the overall performance metrics reflect a significant decline in the model's accuracy when subjected to adversarial examples. While the model maintained a 60% accuracy on the original images, this dropped to 0% on the adversarially perturbed images. Despite this, the model's confidence in its predictions remained relatively high, even when incorrect, with an average confidence of 0.78 on adversarial images compared to 0.84 on original images.

These results underscore the vulnerability of the model to adversarial attacks, which not only cause misclassifications but also induce misplaced confidence in incorrect predictions. The substantial drop in accuracy highlights the need for robust defensive

mechanisms to mitigate the impact of such adversarial perturbations in practical applications.

```
Accuracy on Original Images: 60.00%
Accuracy on Adversarial Images: 0.00%
Average Confidence on Original Images: 0.84
Average Confidence on Adversarial Images: 0.78
```

Figure 23: Overall results from the prediction of ResNet50

Chapter V

5 Discussion

5.1 Metric results from Model Training and Evaluation

In the table below, are the metric results from training the DL models on the CIFAR10 dataset. Because of callbacks (EarlyStopping and ReduceLRonPlateau) the number of epochs differ.

	Accuracy	Loss	Precision	Recall	F1	AUC	Epochs
CNN	97%	43%	88%	85%	86%	98%	100
MobileNetV2	98%	41%	90%	89%	90%	98%	36
ResNet50	97%	56%	87%	84%	85%	97%	20

Table 4: Metric Results from training on CIFAR10 dataset

The metric results from the training on the CIFAR-10 dataset, as shown in Table 4, provide a clear comparison between three different models: CNN, MobileNetV2, and ResNet50. MobileNetV2 stands out with the highest accuracy of 98% and the lowest loss of 41%, suggesting it is the most efficient model in terms of both learning and generalization on this dataset. The model also achieved the highest F1 score and recall, indicating its superior ability to balance precision and recall, thus making it particularly effective at correctly classifying both positive and negative cases.

The CNN model, while achieving a slightly lower accuracy of 97%, maintained a strong performance overall with a balanced precision and recall, leading to a solid F1 score of 86%. Its AUC of 98% matches that of MobileNetV2, reinforcing the model's robustness in distinguishing between different classes. However, the loss for CNN is slightly higher than MobileNetV2, which may indicate more variance in the model's predictions.

ResNet50, despite achieving the same accuracy as the CNN model, shows a higher loss value of 56%, which could imply that the model had more difficulty converging or overfitting issues. Its F1 score and recall are slightly lower than those of MobileNetV2, though still competitive. With an AUC of 97%, ResNet50 demonstrates solid performance but does not quite reach the efficiency seen in MobileNetV2, particularly considering that ResNet50 was trained for fewer epochs (20) compared to the other models.

These results indicate that while all models performed well on the CIFAR-10 dataset, MobileNetV2 consistently outperformed the others across most metrics, making it the most effective model in this comparison.

5.2 Difficulties with Input size

Using deep learning models such as MobileNetV2 and ResNet50 is a unique challenge when working with the 32x32 pixel images in the CIFAR-10 dataset. In order to properly use their architecture, these models, which are usually made to handle higher-resolution pictures, frequently need input sizes of 96x96 pixels or greater. The CIFAR-10 pictures' very small size restricts the spatial information and detail that can be recorded, which can make it more difficult for the models to extract complex characteristics that are necessary for precise categorization.

The models may perform less than optimally when using MobileNetV2 or ResNet50 with 32x32 pictures because of the smaller input size. Larger picture datasets have allowed these models to be refined, allowing their deep layers to identify complex patterns and characteristics. Without any adjustments, applying these models to smaller pictures might lead to a considerable loss of information since the essential traits on which these models rely could be completely or severely compressed, making them less useful for jobs requiring high recall and precision.

The Conv2DTranspose layer is used to upscale the CIFAR-10 pictures from 32x32 to 96x96 pixels in order to resolve this problem. By effectively increasing the input photos' resolution, this layer improves their compatibility with MobileNetV2 and ResNet50. Conv2DTranspose expands the spatial dimensions of the pictures so that the models may run on input sizes that they are intended for. This improves the models' performance on the dataset overall and increases their capacity to extract detailed features. Despite the initial constraints of the CIFAR-10 picture size, our method guarantees that the models may more effectively exploit their deep architecture, producing more accurate and trustworthy classification results.

5.3 Future work

Future work could focus on exploring more advanced adversarial attack methods, such as Projected Gradient Descent (PGD) and Carlini & Wagner (C&W) attacks, to further test the robustness of MobileNetV2 and ResNet50. Additionally, implementing adversarial training as a defense mechanism could be investigated to enhance model resilience against such attacks. This would involve augmenting the training dataset with adversarial examples and comparing its effectiveness across different architectures. Extending this study to higher-resolution datasets like ImageNet or CIFAR-100 would also be valuable in understanding how image resolution impacts model vulnerability to adversarial attacks, thereby broadening the applicability of the findings.

In further testing, a key objective would be to reduce the training loss, as a lower training loss typically indicates that the model is better fitting the data and capturing the underlying patterns more effectively. By fine-tuning hyperparameters, experimenting with different architectures, and employing techniques like regularization or learning rate adjustments, we aim to minimize training loss, which could lead to improved generalization and robustness, especially when the model is confronted with adversarial examples or more complex datasets.

Chapter VI

6 Conclusion

This thesis explores the robustness of deep learning models against adversarial attacks, specifically focusing on their application in image classification tasks using the CIFAR-10 dataset. Through the development and evaluation of various models, including a custom CNN, MobileNetV2, and ResNet50, the study highlights both the strengths and vulnerabilities of these architectures when subjected to adversarial manipulations.

The results demonstrated that while all models achieved high accuracy on clean data, their performance significantly degraded when adversarial examples were introduced. This underscores the pressing need for enhanced defence mechanisms within deep learning frameworks, particularly as AI systems continue to be integrated into critical applications. The use of adversarial training showed promise in improving model robustness, but it also revealed the complexity and computational demands associated with defending against such sophisticated attacks.

Furthermore, the thesis identified that scaling up image sizes to meet the input requirements of more complex models like MobileNetV2 and ResNet50 introduced additional challenges, such as increased computational load and potential overfitting. The application of Conv2DTranspose layers effectively addressed the need for upscaling images, allowing the models to leverage their full potential in feature extraction and classification accuracy.

In conclusion, the findings of this research contribute valuable insights into the ongoing efforts to create more resilient AI systems. As adversarial attacks continue to evolve, it becomes increasingly crucial to refine existing models and explore new techniques that can safeguard AI applications from potential threats. Future work could involve experimenting with alternative defence strategies, applying these models to more diverse datasets, and further optimizing training processes to minimize loss and improve overall model stability in adversarial settings.

7 Bibliography

Al Jaberi, S. M., Patel, A., & Al-Masri, A. N. (2023). Object tracking and detection techniques under GANN threats: A systemic review [Review]. *Applied Soft Computing*, *139*, Article 110224. https://doi.org/10.1016/j.asoc.2023.110224

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Delua, J. (2021). *Supervised vs. Unsupervised Learning: What's the Difference?* Retrieved March 12, 2021 from https://www.ibm.com/blog/supervised-vs-unsupervised-learning/

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

Jia, W., Lu, Z., Zhang, H., Liu, Z., Wang, J., & Qu, G. (2022). Fooling the Eyes of Autonomous Vehicles: Robust Physical Adversarial Examples Against Traffic Sign Recognition Systems. 29th Annual Network and Distributed System Security Symposium, NDSS 2022,

Keras. (n.d.-a). *Keras documentation: Keras Applications.* https://keras.io/api/applications/

Keras. (n.d.-b). *MobileNetV2 function*. https://keras.io/api/applications/mobilenet/#mobilenetv2-function

Keras. (n.d.-c). *Resnet50 function*. https://keras.io/api/applications/resnet/#resnet50-function

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering-A tertiary study [Review]. *Information and Software Technology*, *52*(8), 792-805. https://doi.org/10.1016/j.infsof.2010.03.006

Krizhevsky, A. (2012). Learning Multiple Layers of Features from Tiny Images. *University of Toronto*.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). *Imagenet classification with deep convolutional neural networks*.

Kwon, H. (2023). Toward Selective Adversarial Attack for Gait Recognition Systems Based on Deep Neural Network. *IEICE Trans. Inf. Syst.*, *106*, 262-266.

Kwon, H., & Kim, S. (2022). Restricted-Area Adversarial Example Attack for Image Captioning Model. *Wireless Communications and Mobile Computing*, *2022*, 1-9. https://doi.org/10.1155/2022/9962972

Kwon, H., Kim, Y., Yoon, H., & Choi, D. (2018, 29-31 Oct. 2018). Fooling a Neural Network in Military Environments: Random Untargeted Adversarial Example. MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM),

Kwon, H., & Lee, J. (2021). Diversity Adversarial Training against Adversarial Attack on Deep Neural Networks. *Symmetry*, *13*(3), 428. https://www.mdpi.com/2073-8994/13/3/428

Kwon, H., Yoon, H., & Choi, D. (2019). *Priority Adversarial Example in Evasion Attack on Multiple Deep Neural Networks*. https://doi.org/10.1109/ICAIIC.2019.8669034

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, *521*, 436-444. https://doi.org/10.1038/nature14539

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083. Retrieved June 01, 2017, from https://ui.adsabs.harvard.edu/abs/2017arXiv170606083M

Muncsan, T., & Kiss, A. (2021, 2021//). Transferability of Fast Gradient Sign Method. Intelligent Systems and Applications, Cham.

Naqvi, S. M. A., Shabaz, M., Khan, M. A., & Hassan, S. I. (2023). Adversarial Attacks on Visual Objects Using the Fast Gradient Sign Method. *Journal of Grid Computing*, *21*(4), 52. https://doi.org/10.1007/s10723-023-09684-9

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359. https://doi.org/10.1109/TKDE.2009.191

Peters, M. D. J., Godfrey, C. M., Khalil, H., McInerney, P., Parker, D., & Soares, C. B. (2015). Guidance for conducting systematic scoping reviews. *Int J Evid Based Healthc*, *13*(3), 141-146. https://doi.org/10.1097/XEB.0000000000000050

Ren, K., Zheng, T., Qin, Z., & Liu, X. (2020). Adversarial Attacks and Defenses in Deep Learning. *Engineering*, *6*(3), 346-360. https://doi.org/https://doi.org/10.1016/j.eng.2019.12.012

Rosebrock, A. (2024). *OpenCV CV2 Resize Image (cv2.resize) - PyImageSearch*. Retrieved April 2, 2024 from https://pyimagesearch.com/2021/01/20/opencv-resize-image-cv2-resize/

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., & Bernstein, M. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211-252.

Shi, G., Lin, Z., Peng, A., & Zeng, H. (2023). An Enhanced Transferable Adversarial Attack Against Object Detection. Proceedings of the International Joint Conference on Neural Networks,

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Team, I. D. a. A. (2023). *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?* Retrieved July 6, 2023 from https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks/

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. ACM International Conference Proceeding Series,

Woods, W., Chen, J., & Teuscher, C. (2019). Adversarial explanations for understanding image classification decisions and improved neural network robustness. *Nature Machine Intelligence*, *1*. https://doi.org/10.1038/s42256-019-0104-6

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., & Yuille, A. (2017). Adversarial Examples for Semantic Segmentation and Object Detection. Proceedings of the IEEE International Conference on Computer Vision,

Zanddizari, H., Zeinali, B., & Chang, J. M. (2022). Generating Black-Box Adversarial Examples in Sparse Domain [Article]. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *6*(4), 795-804. https://doi.org/10.1109/TETCI.2021.3122467

Zhou, M. (2022, 11-12 Dec. 2022). Research Advanced in Deep Learning Object Detection. 2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS),