Investigating Adversarial Attacks on Neural Networks: Exploring Explainable AI and The Impact of Feature Selection using MATLAB

Anand Samuel Gunti
201687197

Supervised by Dr.Luisa Cutillo and Dr. Michael Croucher (The Mathworks),

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2023

The candidate confirms that the work submitted is his/her own and that appropriate credit has been given where reference has been made to the work of others.

.

UNIVERSITY OF LEEDS

School of Mathematics

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

Academic integrity statement

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes. I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Name: Anand Samuel Gunti

Student ID: 201687197

Abstract

This work undertakes a comprehensive exploration into the dynamics of neural networks, particularly in the context of adversarial attacks and their robustness. Beginning with a foundational understanding of adversarial attacks, the study delves into the complexities of neural network decision-making, specifically focusing on prominent models like GoogleNet, ResNet50, and Xception. By employing Explainable AI techniques such as Grad-CAM and LIME, the study visualizes and interprets the decision-making processes within these networks, revealing areas most susceptible to adversarial attacks and the impact of feature selection.

Building on these insights, the study formulates hypotheses regarding the impact of the distribution of training data, Image pre-processing, and dropout layers. These hypotheses are evaluated and insights are discussed.

One significant revelation from this exploration is the strong influence of training data quality on model performance. Distinct differences in data resolution and source were observed to impact the networks' resilience against adversarial attacks. Specifically, the 'candle' and 'hourglass' classes, sourced from ImageNet, showcased consistent underperformance compared to other classes sourced from Kaggle with higher resolution.

In summary, this dissertation provides a holistic exploration of the impact of adversarial attacks using Explainable AI, and the various factors such as Feature selection influencing their robustness. Finally, it offers valuable insights and directions for future endeavors in the domain.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep learning has emerged as a dominant approach for tasks like image classification, voice recognition, and Natural Language Processing, revolutionizing how we handle information and data. Its influence on how we process information and data has been transformative, shaping the way we interact with technology and enhancing various aspects of our lives.

Deep learning methods have shown remarkable advancements in terms of processing time, scalability, and reliability. These improvements have opened new possibilities for solving real-world challenges that were previously considered impossible to solve.

As a result, deep learning techniques are now widely adopted across multiple industries and sectors. In healthcare, they have facilitated faster and more accurate diagnosis, personalized treatment plans, and drug discovery. Retail companies utilize deep learning for demand forecasting, customer sentiment analysis, and targeted marketing. Governments employ these methods for enhanced security and optimizing public services. In transportation, deep learning aids in autonomous vehicles, traffic management, and route optimization, and in the finance sector it aids mainly in Credit card fraud detection, Credit Risk Assessment, and Algorithmic Trading

The pervasive use of deep learning has undeniably transformed the way we approach problem-solving and decision-making in various domains. With its continuous progress and refinement, the potential applications of deep learning continue to expand, furthering its impact on society and improving quality of life. As technology and research in this field continue to advance, we can expect even more innovative solutions and improvements in numerous areas.

The future of deep learning is promising, with the potential to revolutionize industries that have yet to fully harness its capabilities. As researchers and industry professionals push the boundaries of this technology, its transformative impact on our daily lives continues to grow stronger, paving the way for an even more efficient and intelligent. However, it is critical to address concerns and potential obstacles that may develop with the broad adoption of deep learning to ensure that its benefits are dispersed evenly and ethically. By doing so, we can

optimize deep learning's potential to have a good and long-term impact on society. One such concern is Adversarial attacks

1.1 Problem Statement

In this dissertation we will be focusing on one of the major concerns about deep learning in recent times, particularly neural networks and its susceptibility to adversarial attacks. Adversarial attacks are defined as follows:

> "Adversarial attacks are deliberate manipulations of input data that look unnoticeable to humans but can cause the neural network to produce incorrect outputs. "
> (Goodfellow et.al , 2015)

let's consider an example of an autonomous car to understand the implications of adversarial attacks on deep learning models.

Imagine an autonomous car equipped with advanced deep-learning algorithms to navigate traffic and safely reach its destination. The car relies on a neural network to process sensor data from cameras and other sensors to perceive the environment and make real-time driving decisions. Suppose an attacker with malicious intent wants to compromise the autonomous car's safety by causing it to make dangerous decisions.Let's see how the attacker can achieve this with ease.

Assume an attacker produces an adversarial speed limit sign that seems to humans to be similar to a real speed limitation sign but contains small modifications. These perturbations are designed to cause the neural network to misinterpret the sign as a stop sign. When the autonomous vehicle comes across an adversarial speed restriction sign on its journey, the neural network interprets it as a stop sign. As a result, the car incorrectly engages the brakes and comes to rest on a roadway where stopping is not permitted. This quick deceleration may result in rear-end crashes with vehicles following closely behind or risky traffic conditions, endangering both the car's passengers and other road users.

The goal of this work is to address the challenge of adversarial attacks on deep learning models such as image classification models and to enhance their robustness and reliability in the face of adversarial manipulation. The work aims to understand the underlying causes of vulnerability to adversarial attacks, assess the impact of such attacks on model performance using Explainable AI, and based on this propose a hypothesis to mitigate the risks posed by adversarial examples. To accomplish this, we will carefully select and analyse various image classification models, both traditional and state-of-the-art deep learning architectures, to understand their strengths and weaknesses in the face of adversarial perturbations.

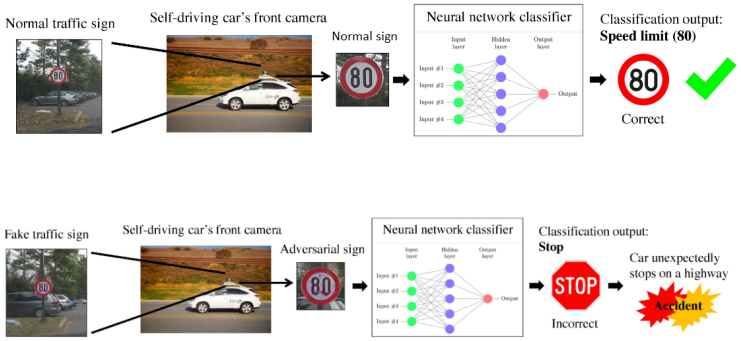

Figure 1.1: illustrating the Adversarial attack example of a self-driving car. (Sitawarin et.al, 2019)

1.2 Objectives of the work

- To gain an understanding of adversarial attacks and their impact on neural networks.
- To investigate how neural networks are fooled by adversarial attacks by using Explainable AI (XAI).
- To formulate a hypothesis or propose a method to enhance the robustness of neural networks against adversarial attacks.

To delve deeper into the specifics of this project, including the full code and associated datasets, please visit my **GitHub Repository.**

The MATLAB code used is available here : `https://github.com/anandgunti/Adversarial-Attacks/tree/main/Programs`

Chapter 2

Background

In this chapter, we will delve into the realm of image classification models and their significance in the context of investigating adversarial attacks. We will provide an overview of the key. concepts and terminologies related to image classification and adversarial attacks.

2.1 Image Classification

Image classification is a fundamental problem in computer vision that involves the classification of an input image into predefined classes or categories. Deep learning algorithms used in image classification models learn to identify patterns and features in images in order to make accurate predictions about the contents of the input image. These models play a crucial role in various real-world applications, such as autonomous vehicles, Facial Recognition, Medical Imaging, Security Surveillance, and e-commerce.

2.1.1 Supervised Learning in Image Classification

Supervised learning is the most common approach in image classification, where the model is trained on labelled datasets, meaning each input image is associated with a corresponding class label. During training, the model learns to map input images to their respective labels through optimization techniques like backpropagation and gradient descent. Convolutional Neural Networks (CNNs) are widely used in supervised learning for their ability to automatically extract relevant features from images and achieve high classification accuracy.

2.1.2 Unsupervised Learning in Image Classification

In contrast to supervised learning, unsupervised learning does not rely on labelled data. Instead, the model attempts to learn patterns and structures directly from the input data without any labels. Autoencoders and Generative Adversarial Networks (GANs) are popular unsupervised learning techniques in the context of image processing. In this work, we are going to use supervised Learning in Image classification. We will discuss this further in this chapter.

2.2 Key Concepts in Image Classification

2.2.1 Neural Network

Biological neurons and artificial neurons in deep learning share fundamental concepts. Each receives inputs, combines them with weighted connections, applies activation functions, and sends out outputs to other neurons. They display complex information processing, hierarchical organization, and adaptability. Artificial neurons simplify these concepts for use in deep learning models, enabling pattern recognition and feature extraction in a variety of tasks. While biological neurons are a crucial component of complex nervous systems, artificial neurons are not a part of these systems. To develop effective machine learning algorithms, biology has been a major source of inspiration.

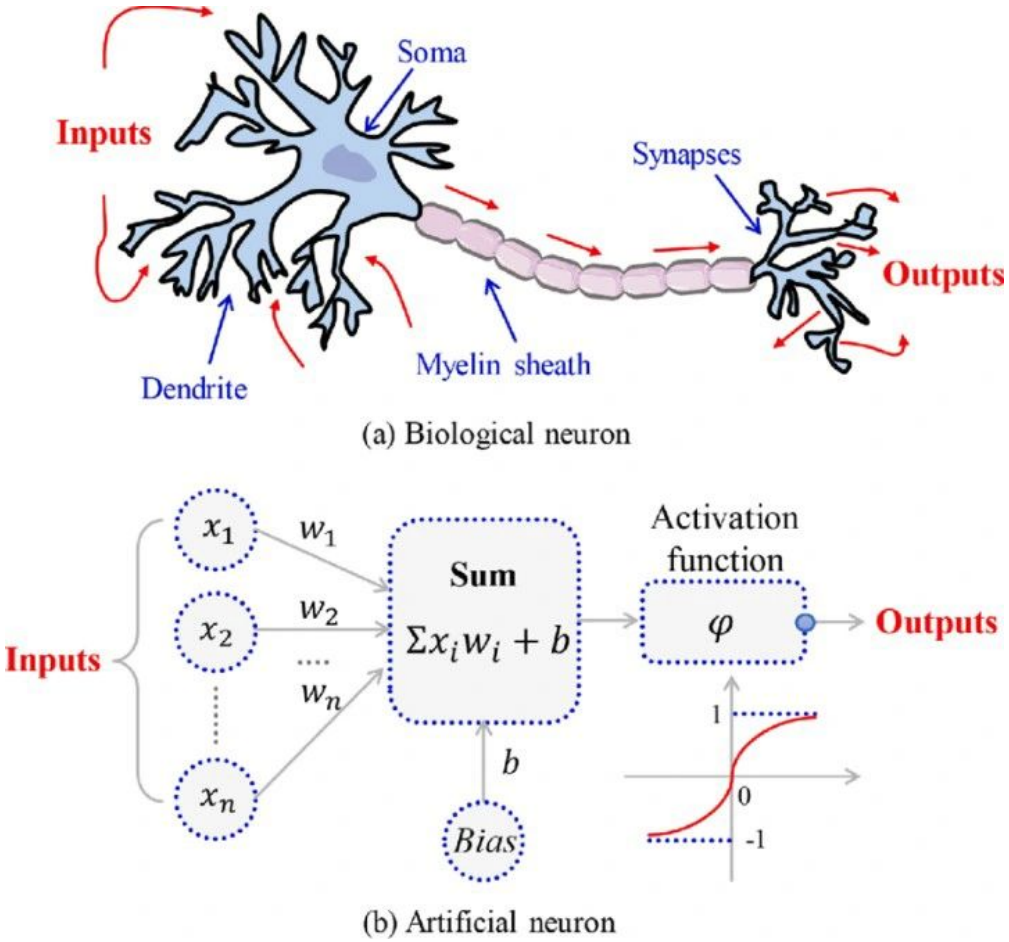

Figure 2.1: Illustration of similarities between Biological and Artificial neuron (Andrej K, 2015)

Neural networks are a collection of algorithms that are intended to recognize patterns and are loosely based on the human brain. They categorize or group raw input to understand sensory data using a form of machine perception. All real-world data, including images, sounds, texts, and time series, must be converted into vectors for them to recognize the patterns, which are

numerical and contained therein. (Nicholson, 2023).

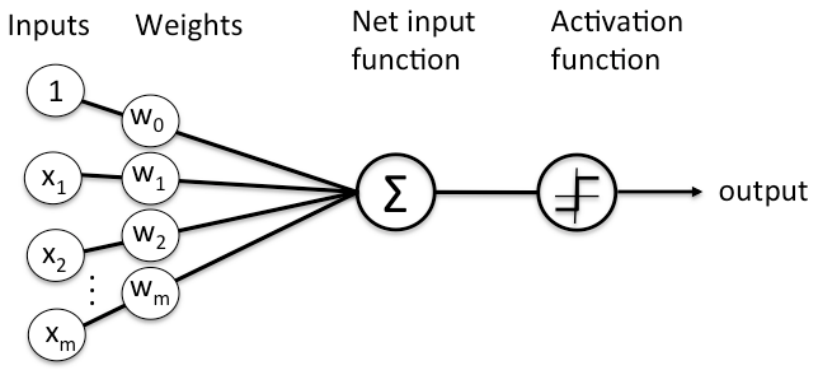

Figure 2.2: illustration of neural network elements in a single neuron with "m" inputs (Nicholson, 2023).

2.2.2 Multilayer Perceptron Neural Networks

Multilayer perceptron (MLPs), often referred to as feedforward neural networks or Deep feedforward networks, represent a fundamental type of deep learning model. It's an extension of the original single-neuron model and consists of multiple layers, allowing it to capture more complex patterns in data. Examples include spam filtering, speech recognition, etc.

The training process of an MLP involves two main steps: forward propagation and backward propagation. During forward propagation, the input data is passed through the network to calculate the activations of the hidden layers and the output layer. The outputs are obtained based on the inputs and the current weights. During backward propagation, the algorithm adjusts the weights and biases of the network based on the error of the prediction.

2.2.3 Forward Propagation Architecture

Input Layer: The input layer is the initial layer of a neural network. It's where the raw data or features are introduced to the network for processing. Each neuron in the input layer corresponds to a specific input feature, and the number of neurons in the input layer is determined by the dimensionality of the input data. For example, if the size of the input image is 224 * 224 pixels then the input layer will have 50,176 neurons.

The input layer serves as an entry point for the raw data to enter the neural network. It's the first step in multi-layer processing, where the data is passed through hidden layers to ultimately produce an output enabling the network to learn and derive meaningful representations from the input data.

Hidden Layer: Hidden layers are the intermediate layers in a neural network. These layers lie between the input layer and the output layer and play a crucial role in extracting hierarchical

features and representations from the input data. Each hidden layer processes the information passed from the previous layer and feeds it to the next layer. if the number of hidden layers is more than one then the network is known as a deep neural network and if there is only one layer it is known as a shallow neural network.

Output Layer The output layer is the final layer of a neural network, including multi-layer perceptrons (MLPs) and other deep learning architectures. It produces the network's final predictions or outputs based on the processed input data and the learned patterns in the hidden layers.

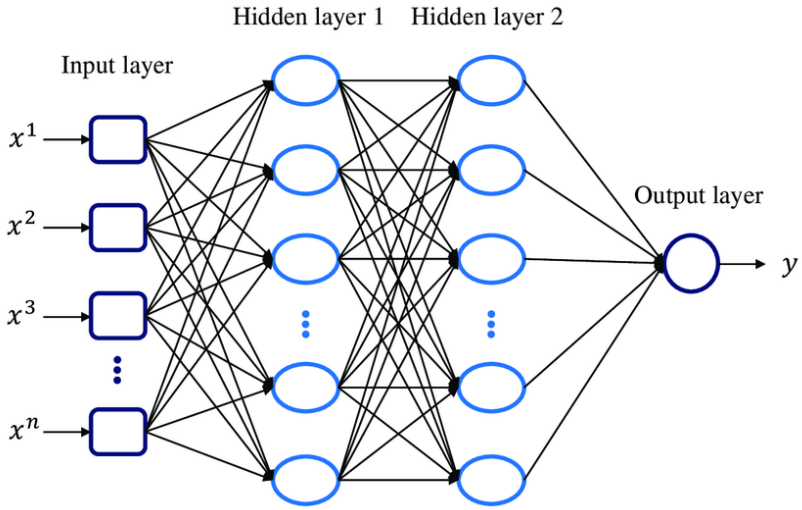

Figure 2.3: Multilayer perceptron (MLP) architecture example with two hidden layers and one prediction output (Alireza Sarraf Shirazi & Ian Frigaard , 2021)

2.2.4 Loss Function

A loss function, also known as a cost function or objective function, is an important component in training machine learning models, like multi-layer perceptrons (MLPs) and Convolution neural networks (explained in 2.3). It quantifies the difference between the model's predictions and the actual values, providing a measure of how well the model is performing on a given task. Our goal throughout the process will be to keep this as minimum s possible. This can be achieved by using optimizers such as Stochastic Gradient Descent (SGD), Adam commonly used in image classification tasks

Commonly used Loss functions in classification tasks

Softmax Loss (Multinomial Logistic Loss): Softmax Loss is used for multi-class classification problems. The Softmax function turns logits (numeric output of the last fully connected layer of a multi-class classification neural network) into probabilities for each class. The loss is then computed by taking the negative log-likelihood of the true label. (stanford CS231 , 2023)

Cross-Entropy Loss (Log Loss): Cross-Entropy Loss is used for classification problems, especially when the outputs are probabilities It measures the performance of a classification model whose output is a probability value between 0 and 1. we can use Softmax for probabilities and then apply Cross-Entropy Loss. (Goodfellow et.al , 2016) The formula for Cross-Entropy Loss is :

$$CrossEntropy(y, \hat{y}) = -\sum_{i} y_i \log(\hat{y}_i) \tag{2.1}$$

The magnitude of the loss function gives us an idea about the performance of the model in terms of how far off are the model predictions from true values.it is one of the hyperparameters which can be tuned to improve the performance of the model.

2.2.5 Back propagation

Back propagation enables the network to adjust its weights and biases in a way that minimizes loss using a specified loss function, ultimately improving its ability to make accurate predictions on an iterative basis. Backpropagation plays a crucial role in improving the model's ability to capture complex relationships in data and make accurate predictions.

2.2.6 Activation Function

Generally, in Machine learning /deep learning, we have broadly two different types of models based on the relationship between input and output features. For simple relationships, we use linear models and for complex relationships, we use non-linear models.

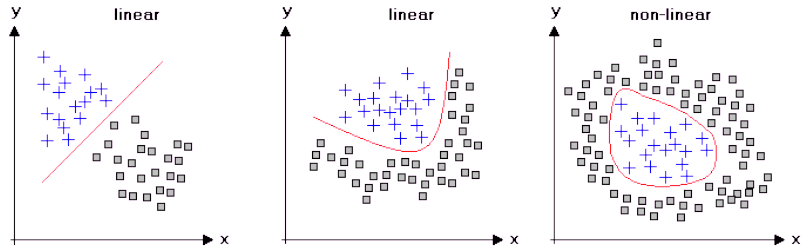

Figure 2.4: : Linear and Non-linear models Representation (H. Lohninger , 1999)

For linear models, we don't require activation functions as the input is linear and the output will be a simple linear function and easy to solve as they can directly compute a weighted sum of inputs to predict the output. However, real-world problems such as Image Classification, Voice Recognition, Image Captioning, etc. involve complex and nonlinear relationships. To capture these intricacies, we require activation functions as they induce non-linearity in our neural networks. This is the reason that we use activation functions to help with complicated,

high dimensional, and nonlinear datasets . (Sharma et.al, , 2020)

Activation functions, like ReLU, sigmoid, or tanh, introduce the nonlinear element in neural networks. They take the weighted sum of inputs and apply a mathematical transformation that introduces curves and fits the data. This allows the network to learn and represent complex decision boundaries, which is critical for solving problems involving intricate data patterns.

2.3 Convolution Neural Networks

Convolutional Neural Networks (CNNs) are a class of supervised deep neural network architectures designed primarily for processing and analysing visual data, such as images and videos. CNNs are inspired by the human visual system, as they are particularly adept at capturing relationships and patterns in data. Their ability to learn and extract meaningful features from visual data makes them versatile in the Technological field. CNN's important applications include image classification, image segmentation, object detection, Medical imaging, and Autonomous vehicles. We examined and implemented CNN models, specifically GoogLeNet, ResNet50, and Xception, and discussed them in Chapter 5.

2.3.1 Key Concepts in CNN:

- **Convolutional Layers:** These layers perform convolution operations by sliding a set of filters (also known as kernels) over the input image or feature map. Each filter extracts specific features from the input data, such as edges, textures, or more complex patterns. The convolution operation produces feature maps that highlight relevant features in the data.

- **Filter/Kernel:** A small matrix that is convolved with the input image to perform operations like edge detection, blurring, or feature extraction.

- **Flattening**: Before passing the output of convolutional and pooling layers to fully connected layers, the feature maps are flattened into a one-dimensional vector.

- **Stride:** The stride parameter in convolutional operations determines how much the filter moves after each convolution. A larger stride reduces the spatial dimensions of the feature map.

- **Padding:** Padding involves adding additional rows and columns to the input image or feature map before performing convolution. It helps maintain the spatial dimensions of the output feature map, preventing them from becoming smaller.

- **Transfer Learning:** Transfer learning involves using pre-trained CNN models as a starting point and fine-tuning them for specific tasks. we will discuss about this in chapter 6

- **Dropout:** A regularization technique where random neurons are temporarily ”dropped out” during training to prevent over-fitting.

2.4 Model performance metrics

Model evaluation is crucial in machine learning to understand how well a model is performing and where it might need improvement. While there are various metrics available, in this work we will be using accuracy and F1 score as they are among the most widely used due to their interpretability and relevance.

2.4.1 Accuracy

Accuracy measures the fraction of all predictions that are correct. It's a general metric that provides a quick snapshot of overall model performance.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2.2}$$

Insights:

- A high accuracy indicates that the model, in general, performs well in making correct predictions.
- However, accuracy can be misleading in the presence of class imbalances where one class significantly outnumbers others.

2.4.2 F1 Score: Balancing Precision and Recall

The F1 score is the harmonic mean of precision and recall, making it a comprehensive metric that accounts for both false positives and false negatives.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.3}$$

Insights:

- An F1 score provides a balance between the model's ability to correctly identify positive instances (recall) and its ability to ensure that a positive prediction is indeed positive (precision).
- It's particularly useful in scenarios where false positives and false negatives have different costs or where class distributions are skewed.

Chapter 3

Literature Review

3.1 Intriguing properties of neural networks - Szegedy et.al, (2014)

This research paper is crucial for understanding the properties of neural networks. This paper investigates the phenomenon of adversarial examples and highlights the surprising vulnerabilities of neural networks to small, engineered perturbations to input data. The authors discussed two counter-intuitive properties of deep neural networks, which are powerful learning models used in tasks such as image recognition.

Previous research has led us to believe that the specific patterns or features that the network has learned to identify in neural networks known as semantic information are stored within the individual activations of its components. This understanding assumes that the components in the final layer hold significance in extracting meaningful insights from the data. In other words, these works aim to identify the patterns in the input data that lead to the strongest response from a particular neuron. But this idea/theory was invalidated by this paper, and it proved that it is the entire space of activations, rather than the individual units, that contains the bulk of the semantic information. This is the first property that the authors had identified.

The second counter-intuitive property is the identification of the existence of Adversarial examples.

> "Consider a state-of-the-art deep neural network that generalizes well on an object recognition task. We expect such network to be robust to small perturbations of its input, because small perturbation cannot change the object category of an image. However, we find that applying an imperceptible non-random perturbation to a test image, it is possible to arbitrarily change the network's prediction. These perturbations are found by optimizing the input to maximize the prediction error. We term the so perturbed examples adversarial examples." ((Szegedy et.al , 2014)).

The authors have demonstrated that if we make small changes to input that are imperceptible these changes can mislead the network. This is an instance where the vulnerability of models

in this direction has been explored and also the basis for research on adversarial attacks on neural networks. The authors further inferred from their experimentation about cross-model generalization that an attack created for a particular network can also fool other neural networks that have different architectures (number of layers and initial weights & biases) and we are going to explore this concept in this work by comparing it with different pre-trained models. The authors have also established that these adversarial examples are not case-specific, but they can be generalized to the entire dataset.

The insights from this paper laid the foundation for the research of adversarial attacks. By explaining that the behaviour of neural networks is based on the activations of the space, the paper sets the stage for further investigations into vulnerabilities posed by adversarial examples.

3.2 Explaining and Harnessing Adversarial Examples – Goodfellow et.al, (2015)

The previous work by (Szegedy et.al , 2014) just gave us a glimpse of adversarial examples but this paper has extensively explained the adversarial examples and the reasoning behind the generalization of adversarial examples discussed in the above paper. The paper introduces the Fast Gradient Sign Method (FGSM), a technique to generate adversarial examples efficiently.

An adversarial example is created by adding a perturbation (denoted as η) to the original input (x). The goal is to modify the input slightly so that the model's prediction changes. Now the adversarial input $x' = x+\eta$. In terms of the dot product between the weights and adversarial example (x').It is expressed as

$$W^T x' = W^T x + W^T \eta \tag{3.1}$$

In high-dimensional problems (where the input has many features), the effect of a small perturbation η on the activation can be magnified because the adversarial perturbations increase by $W^T \eta$. This is because the linear combination of weights and perturbation can accumulate across multiple dimensions. As the dimensionality of the problem increases, the potential for adversarial examples also increases. From these authors demonstrated that even small perturbations in high-dimensional space can cause significant shifts in the model's predictions.

Based on this the authors established that adversarial examples are based on the high dimensional nature of the input (Linearity) not based on entire space activations (non-linearity) as discussed by (Szegedy et.al , 2014).

The ability of adversarial examples to influence different models, even those with different architectures, can be understood by two factors discussed in this paper. First, the perturbations introduced to create adversarial examples align closely (following the same direction as weights) with the internal weight vectors of a model. This is due to the mathematical relationship between

the perturbations and the weights in the above equation as explained above.

Second, when different models are trained to perform the same task, they tend to learn similar underlying features/patterns. This means that the vulnerabilities introduced by adversarial perturbations in one model are likely to impact other models trained for the same task. We will be investigating this in future chapters.

3.3 Adversarial Examples in The Physical World - Kurakin et.al, (2016)

This paper examines the practical implications of adversarial attacks by demonstrating their effectiveness in real-world scenarios involving image recognition systems until now which are explained in theory by previous papers by giving input directly to the model. This paper explores the concept of reconstructing the adversarial images physically and see whether the network still misclassifies it when they are captured by the camera. The authors have successfully demonstrated this.

The paper also discusses the ways to generate adversarial examples and the authors introduced the Basic Iterative method to extend the Fast Gradient Sign Method (FGSM). The authors applied the FGSM multiple times with a small step size known as alpha. These Iterative methods aim to make minimal changes at each step (alpha) while still causing misclassification.

The authors have also provided insights into transformations of the images and their effect on the robustness of the model.

> "We found that "fast" adversarial images are more robust to photo transformation compared to iterative methods. This could be explained by the fact that iterative methods exploit more subtle kind of perturbations, and these subtle perturbations are more likely to be destroyed by photo transformation." (kurakin et.al , 2016)

In other words, when an original image is perturbed to create an adversarial example using the fast method, and the resulting perturbed image is subjected to changes in lighting, rotation, scale, or other types of alterations that can occur in the real world referred to as "photo transformations", the adversarial nature of the perturbed image remains relatively intact. On the other hand, when adversarial examples are generated using iterative methods, the subtle perturbations introduced might be more sensitive to these photo transformations. This sensitivity could lead to the adversarial effect being diminished or even lost when the perturbed image is transformed.

The authors have also introduced a parameter to study the influence of arbitrary transformations on adversarial images called **destruction rate.** It is a measure of the number of adversarial images that are no longer misclassified after undergoing arbitrary transformations. The destruction rate is calculated using a formula that considers the number of images, the true class of the image, the corresponding adversarial image, and the image transformation function.

3.4 Adversarial Examples Are Not Bugs, They Are Features – (Ilyas et.al , 2019)

Previous work has proposed various explanations for the reason behind the network's misclassification of adversarial examples, but the reasons for their existence and pervasiveness remain unclear. In this paper, the authors propose a new perspective on the phenomenon of adversarial examples, explaining that adversarial vulnerability is a fundamental consequence of sophisticated learning algorithms.

The authors demonstrated theoretically that adversarial examples can be directly attributed to the presence of non-robust features. These features are highly predictive features that are present in the data distribution but are brittle and hard to understand for humans. These features are often exploited by machine learning models to achieve high accuracy on natural inputs, but they are also the reason why models are vulnerable to adversarial examples. They are not robust to small perturbations, small changes, and cause the model to make incorrect predictions.

We would discuss non-robust features with an example let's consider we have a dataset of cat images, and some of these images have a distinctive watermark in the Top corner. While the watermark itself doesn't have anything to do with whether an image contains a cat or a dog, a neural network could mistakenly associate the presence of the watermark with one of the classes if it appears more frequently on a specific class. In this case, the presence of the watermark is a non-robust feature because the presence or absence of the watermark might lead to incorrect predictions. The model might not generalize well to new images that don't have the same watermark.

Insights

- The authors have also provided an insight into the transferability of the adversarial examples they have established that it is due to the non-robust features of the data.
- They have also provided the Theoretical Framework for Studying (Non)-Robust Features.
- The paper suggests that adversarial robustness can be improved by maximizing accuracy, either through improved standard regularization methods or image pre-processing.
- The authors established that non-robust features can be picked up by models during standard training, even in the presence of robust features that are predictive.

3.5 Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models" - Brendel et al, (2018).

Until now we are discussing adversarial attacks, but we are generalizing them. This paper discusses Adversarial attacks in detail and categorizes them. The paper emphasizes the importance

of decision-based attacks, which solely rely on the final model decision, and introduces the Boundary Attack, a decision-based attack.

Adversarial attacks are divided into three categories: gradient-based, score-based, transfer-based, and attacks.

Gradient-based attacks: These attacks depend on model information including the gradient of the loss function. Examples are the Fast-Gradient Sign Method (FGSM) (Goodfellow et.al, 2015), the Basic Iterative Method (BIM) (Kurakin et al., 2016), DeepFool (Dezfooli et al., 2015), the Jacobian-based Saliency Map Attack (JSMA) (Papernot et al., 2015), Houdini (Cisse et al., 2017) and the Carlini & Wagner attack (Carlini & Wagner, 2017). Masking the gradients is an efficient way to defend against gradient-based attacks.

Score-based attacks: These attacks are completely dependent on the predicted scores of the algorithm. These attacks use the predicted probabilities to numerically calculate the gradients of the loss function. Examples of score-based attacks include black-box variants of JSMA (Narodytska & Kasiviswanathan, 2016) and the Carlini & Wagner attack (Chen et al., 2017). The defence for these types of attacks is to impede the calculations of the gradients and this can be done by using the dropout in the models.

Transfer-based attacks: These attacks require information about the training data only. These surrogate the existing model and from that these can generate adversarial examples (Papernot et al., 2017). Defences against transfer-based attacks include robust training on a dataset augmented by adversarial examples.

> "The fact that many attacks can be easily averted makes it often extremely difficult to assess whether a model is truly robust or whether the attacks are just too weak, which has led to premature claims of robustness for DNNs (Carlini Wagner, 2017; Brendel Bethge, 2017). This motivates us to focus on a category of adversarial attacks that have so far received fairly little attention: decision-based attacks. " (Brendel et.al , 2018)

Decision-based attacks: These attacks solely rely on the final decision of the model and are more relevant in real-world applications. These are more robust to standard defences compared to score-based attacks or transfer-based attacks, as they require less information about the model and are simpler to apply. These are also proven to be efficient against gradient masking.

One of the main contributions of this paper is the introduction of an effective decision-based attack that can fool the complex models known as The Boundary Attack. The intuition behind the Boundary attack as explained by the authors is that the algorithm starts from an adversarial point and moves along the boundary between the adversarial and non-adversarial regions, reducing the distance to the target image. The authors have tested this attack type against the defence methods discussed above and have also drawn a comparison between the

attack types. The results have shown that The Boundary attack is more robust to the defence methods.

3.6 One Pixel Attack for Fooling Deep Neural Networks - Jiawei Su et.al (2019)

We are discussing the adversarial attacks in the previous papers and how to generate them. But what is the slightest change you need to make a network misclassify is something we are wondering, and this paper gave us the answer that it is one pixel that needs to be changed emphasizing the subtlety and potency of adversarial attacks. Now here the question arises which pixel is to be selected. The authors used the concept of differential evolution (DE) optimization algorithm for selecting the pixel. The reason behind using the differential evolution is the Higher probability of Finding Global Optima, Require Less Information from Target System, and simplicity as it is independent of the classifier used.

Differential evolution (DE) optimization algorithm is used to generate candidate solutions iteratively, where each solution represents a perturbation that modifies one pixel. In the context of the one-pixel attack, each candidate solution represents a perturbation that modifies one pixel. The candidate solutions are encoded into an array, with each perturbation represented by five elements: x-y coordinates and RGB value as specified by the authors. Then these solutions are compared against a fitness function which in this case the probabilistic values of the targeted class and true class. The DE algorithm aims to find the optimal pixel perturbation that maximizes the probability of the target class label while minimizing the perturbation strength.

3.7 Summary

This literature review is the most crucial part of this work as further objectives were derived from this. We got a basic understanding of the intuitive properties of deep neural networks and their behaviour from (Szegedy et.al , 2014). This is the fundamental paper that identified the existence of adversarial attacks. (Goodfellow et.al , 2015) has built on the previous work and explained the reasoning behind the adversarial examples by countering the explanation provided by previous researchers. The authors have also introduced an efficient way to generate adversarial examples known as FGSM and provided explanations for the transferability of the attacks.(kurakin et.al , 2016) have explored the practical implications by generating real-world examples, and in the process, the authors have introduced the Basic Iterative method for generating Adversarial examples. (Ilyas et.al , 2019) have given more meaningful and plausible explanations behind the adversarial attacks and their transferability across the models attributing it to the non-robust features of the network. We have also discussed the minimal change required to generate an adversarial attack i.e., single pixel attack (Su et.al , 2019).

We have a solid grasp of neural network behaviour and the generation of adversarial examples, and the theories behind them. Visualizing these attacks provides concrete proof of vulnerabilities resulting from even slight changes and gives us insights regarding network decision-making. we can also verify the transferability of the attacks across various models and compare them visually to understand. This can be done by using Explainable AI which we will discuss in the next chapter.

Chapter 4

Methodology

The methodology section of this work outlines the systematic approach undertaken to achieve the objectives of gaining insight into adversarial attacks' impact on neural networks, investigating the mechanisms underlying neural networks' vulnerability to such attacks using explainable AI techniques, and formulating a hypothesis or proposing a novel method to bolster neural network robustness against adversarial perturbations.

The in-depth comprehension of adversarial attacks and their consequences on neural networks, as elucidated in Chapter 3, provides the base for the forthcoming investigation of adversarial attacks using Explainable AI.

4.1 pre-trained networks used for image classification.

The primary step will be to do the image classification of our dataset on pre-trained networks using transfer learning in MATLAB. We will be using 3 pre-trained networks which are mostly used and efficient in computer vision models.

4.1.1 ResNet50

ResNet-50 is a convolutional neural network that is 50 layers deep. ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. ResNet allowed us to train extremely deep neural networks with 150+ layers. It is an innovative neural network that was first introduced in the research paper titled 'Deep Residual Learning for Image Recognition'(He et.al , 2015).

ResNet-50 uses a bottleneck architecture. This means that each block consists of three convolutional layers: a 1x1 convolution for dimension reduction, followed by a 3x3 convolution, and another 1x1 convolution for dimension expansion. This reduces the computational cost while still capturing important features (He et.al , 2015). This advantage makes Resnet a widely used base model for transfer learning.

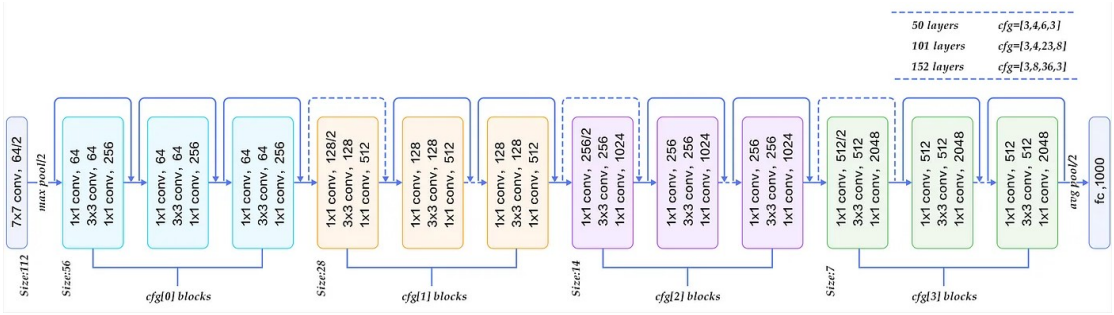

Figure 4.1: Architecture of ResNet (He et.al , 2015)

4.1.2 GoogLeNet

GoogLeNet, is inspired by the LeNet architecture, is a convolutional neural network architecture that consists of 22 deep layers that were introduced in 2014 by researchers at Google in their paper, ”Going Deeper with Convolutions”.This was trained on ImageNet dataset with 1000 classes.(szegedy et.al , 2015)

4.1.3 Xception:

Xception has been trained on a dataset of 14,000 images with dense annotations from about 6,000 classes making it the state of art computer vision model known for its computational efficiency (Chollet , 2017). This was developed by François Chollet and published by him in the research paper titled “Xception: Deep Learning with Depthwise Separable Convolutions” in 2017.

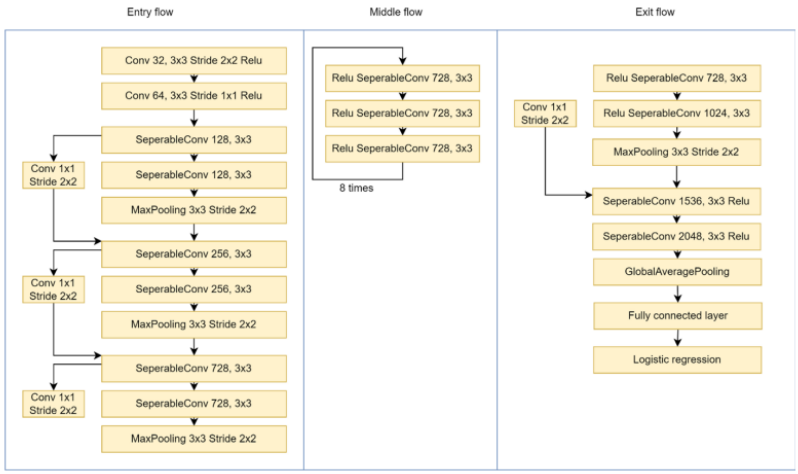

Figure 4.2: Architecture of Xception Model (Gulmez , 2023)

Note: For Resnet and GoogleNet the input size of the image is 224 pixels * 224 pixels and for Xception input size is 299 pixels * 299 pixels.

We will assess our dataset by employing three pre-trained neural network models. Subsequently, we will evaluate the models' performance using an accuracy metric. We will be analysing the predictions of the network using Grad-CAM which will help us in visualizing the behaviour of the network and the reasoning behind its predictions. We will be discussing Grad-CAM further in this chapter.

4.2 Simulating Adversarial Attacks.

Once we have established the baseline performance of our models, we need to simulate the adversarial attacks. In this work, we would be working on Targeted and Untargeted adversarial attacks. Targeted adversarial attacks are specific type of adversarial attack where the attacker intentionally manipulates the input data to force the model to classify it as a specific target class. Unlike non-targeted attacks, which aim to cause misclassification without a specific target in mind. we would be using the FGSM (section 3.2) for untargeted attacks and the Basic Iterative method (section 3.3) for Targeted attacks.

4.2.1 FGSM Method

This method calculates the gradient $(\nabla_x J(\theta, x, y_{\text{true}}))$ of the loss function J, with respect to the image ϵ. we want to find an adversarial example for, and the class label y_{true}. This gradient describes the direction to "push" the image in to increase the chance it is misclassified. we can then add or subtract a small error from each pixel to increase the likelihood the image is misclassified.

The adversarial example is calculated as follows:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y_{\text{true}})) \tag{4.1}$$

- where x' is the adversarial example,
- x is the original input image,
- ϵ is the perturbation magnitude, $\nabla_x J(\theta, x, y_{\text{true}})$ is the gradient of the loss with respect to x, and θ represents the model's parameters.

Parameter ϵ controls the size of the push. A larger ϵ value increases the chance of generating a misclassified image but makes the change in the image more visible. This method is untargeted, as the aim is to get the image misclassified, regardless of which class.

4.2.2 Basic Iterative Method

A simple improvement to FGSM is to perform multiple iterations. This approach is known as the basic iterative method (BIM) (kurakin et.al , 2016). For the BIM, the size of the perturbation

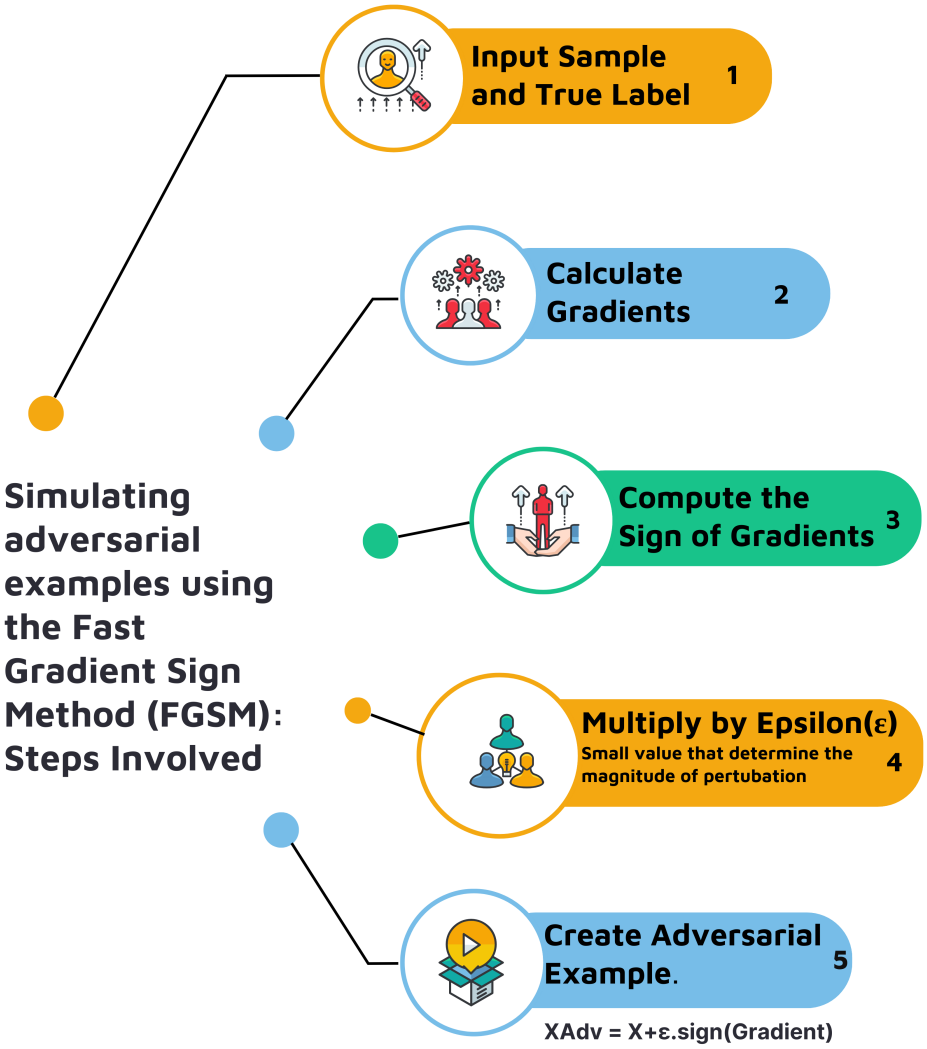

Figure 4.3: Steps involved in simulating FGSM attack.

is controlled by a parameter α representing the step size in each iteration. This is because the BIM usually takes many, smaller, FGSM steps in the direction of the gradient. After each iteration, clip the perturbation to ensure the magnitude does not exceed ϵ. This method can yield adversarial examples with less distortion than FGSM. When you use untargeted FGSM, the predicted label of the adversarial example can be very similar to the label of the original image. For example, a dog might be misclassified as a different kind of dog. However, you can easily modify these methods to misclassify an image as a specific class.

4.3 Explainable AI

Explainable AI (XAI) refers to the set of techniques and methods aimed at making the decisions and outputs of artificial intelligence (AI) systems understandable and transparent to humans. The goal of XAI is to bridge the gap between the ”black box” nature of complex AI models and the need for interpretable explanations for their behavior. (Pingel,MathWorks Blog , 2023).

Explainable Artificial Intelligence (XAI) methods enhance deep learning models by providing explanations for their behavior and decision-making processes. These techniques offer benefits such as building trust, ensuring accountability, promoting transparency, and addressing fairness concerns. (Bennetot et.al , 2022). XAI techniques allow users to understand how the model arrives at its predictions, enabling them to verify and certify the model outputs. By having access to explanations, we can evaluate the model's trustworthiness and hold it accountable for its decisions. Transparency is achieved through the ability to interpret the model's internal workings, making it easier to identify biases and errors. XAI techniques also promote fairness by enabling the identification and mitigation of biases in the model's decision-making process, ensuring that the model does not discriminate against certain groups or individuals. (Bennetot et.al , 2022). Overall, XAI techniques provide a means to understand, evaluate, and regulate machine learning models, making them more reliable, accountable, transparent, and fair.

4.3.1 Grad CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a technique used for image classification that produces a coarse localization map highlighting the important regions in the image for predicting the concept. This technique was proposed in a paper titled “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization” (Ramprasaath R. Selvaraju et .al , 2019)

It utilizes the gradients flowing into the final convolutional layer of a deep learning model to produce a coarse localization map, highlighting the regions that are crucial for predicting a specific concept or class. By computing the gradients of the score for a particular class with respect to the feature map activations, Grad-CAM generates a heatmap that indicates the importance of different regions in the image.

Grad-CAM is easy to understand and implement for any gradient-based model, as it does not require modifying the model architecture. However, the interpretation of the heatmap is subjective and can introduce human bias, as it relies on the user's interpretation of the visual explanation.

Grad-CAM visualizes what regions of an image are essential for a Convolutional Neural Network (CNN) classification. Given an image and a target class, the image is processed through the CNN to get a logit score for that class. Only the desired class's gradient is kept (e.g., 'tiger cat'), while others are zeroed. Backpropagation to convolutional layers yields a coarse heatmap

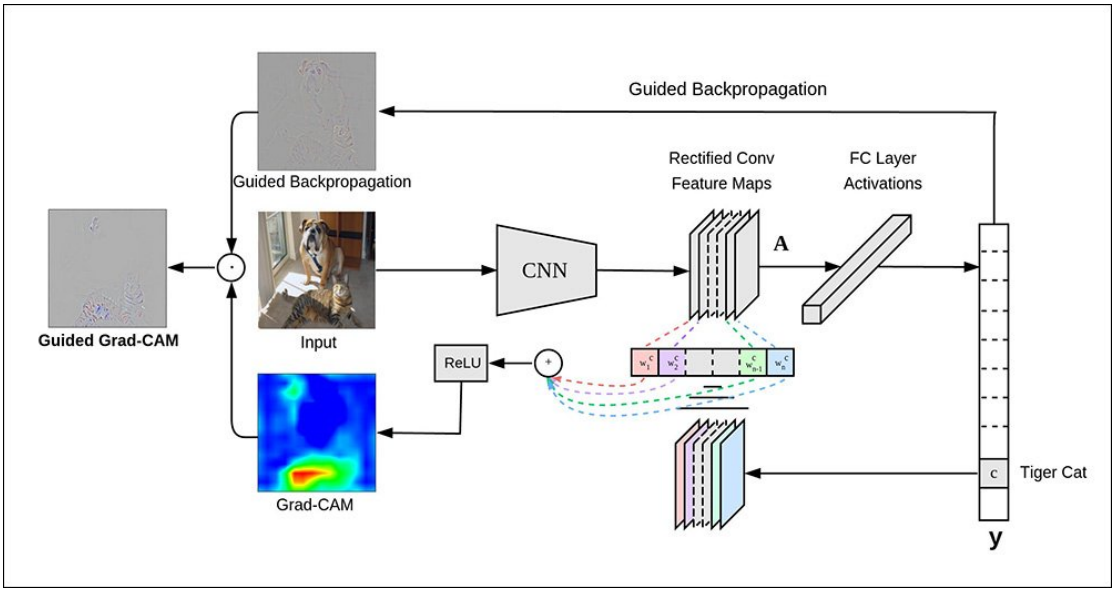

Figure 4.4: The architecture of Grad CAM (Ramprasaath R. Selvaraju et .al , 2019)

(Figure 4.4), showing the decision-making regions. Combining this with guided backpropagation produces Guided Grad-CAM visualizations. These offer high-resolution, class-specific insights into model decisions, revealing where the model looked to classify the image into the specific category.

Mathematical Steps:

These steps are solely based on a paper titled "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization" (Ramprasaath R. Selvaraju et .al , 2019)

1. Let f^k be the activations of a convolutional layer for a given input image, where k is the index of the feature map.

2. Compute the gradient of the score for the class c (before the softmax) with respect to these feature maps:
$$\frac{\partial y^c}{\partial f^k} \tag{4.2}$$
where y^c is the score before softmax for class c.

3. Global-average-pool the gradients over width and height dimensions to obtain the neuron importance weights α_k^c:
$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial f_{ij}^k} \tag{4.3}$$
where Z is the number of pixels, and i, j are pixel indices.

4. Finally, compute the weighted combination of activation maps and pass it through a

ReLU:

$$L^{c}_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_{k} \alpha^{c}_{k} f^{k}\right) \tag{4.4}$$

This results in a 2D spatial heatmap that highlights the regions most influential for class c.

4.3.2 LIME - Local Interpretable Model-Agnostic Explanations

LIME is a explanation technique that aims to provide interpretable explanations for the predictions made by any machine learning model. It aims to address the challenge of understanding complex models, especially in contexts where interpretability is crucial It achevies this by creating a simpler model that approximates the original one, but only close to the specific data point in question (Marco Tulio Ribeiro et.al, 2016). Instead of providing a global understanding of the model on the entire dataset, LIME focuses on explaining the model's prediction for individual instances.
LIME works by creating a interpretable model that approximates the behaviour of the complex model. The simpler model is trained on local data points, and the resulting model is used to explain the decision of the complex model. Simplified steps involved in LIME (Safjan, 2023)

- Selecting an instance to explain
- Generate the instance of the dataset by randomly altering the features of the original dataset instance.
- Use the original model to predict outcomes on the generated instance
- Calculate the similarity between the original and generated instance
- Train the model on the generated instances and use the calculated similarities to approximate the model decision near the original instance
- Using the Trained local model to generate explanations for the original model's decision.

LIME is a powerful tool for XAI that provides local, interpretable explanations for individual predictions made by any deep learning model. LIME can generate explanations that are tailored to specific instances. However, LIME also has some limitations, such as its sensitivity to perturbations.

4.4 Comparative Analysis

After obtaining the results from our pre-trained models, transfer learning models, adversarial attacks, and Explainable AI methods, a comparative analysis will be vital.

- **Adversarial Impact:** Compare the confidence scores of the models' predictions on both original and adversarial images. This will highlight how adversarial perturbations affect the model's certainty.

- **Explainable AI Discrepancies:** Utilize XAI methods to identify if there are patterns or specific regions within images that adversarial attacks consistently target. Comparing the heat maps of original images with adversarial ones will provide insights into these discrepancies.

- **Transfer Learning Resilience:** Examine if transfer learning made the models more or less resilient to adversarial attacks compared to their pre-trained counterparts.

4.5 Evaluation and Hypotheses Testing

We employed explainable AI techniques to visualize the black-box characteristics of the neural networks. Our intention was to extract insights from these visualizations and gain an understanding of how the network responds to adversarial examples in contrast to the original images. we aim to recognize potential visual variances between these instances. Subsequently, we aim to construct hypotheses to explain the rationale behind the network's responses. Based on the hypothesis we will be introducing modifications to evaluate the hypotheses.

Chapter 5

Experimentation and Inferences

5.1 Baseline Performance Using Pre-Trained Models on Single Image Classification

Evaluating a model's prediction on a single image can provide insight into its recognition capability, especially in cases where we want to understand or showcase how a particular model perceives specific image content. In this, we will be using pre-trained models such as GoogLeNet, Resnet50, and Xception. These models are trained on an ImageNet dataset with 1000 classes.

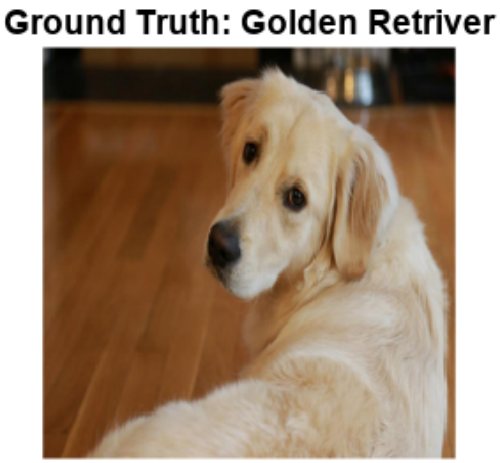

Figure 5.1: Image used to Evaluate the baseline performance

Before passing to the model, we would do the input pre-processing as required per the models individually. Then we would evaluate the model performance based on the confidence score.

Model	Predicted class	Top 3 Confidence Scores
GoogLeNet	Golden Retriever	Golden Retriever (55.42 %) Labrador Retriever (39.63%) Kuvasz (2.54%)
ResNet50	Golden Retriever	Golden Retriever (89.89%) Labrador Retriever (4.09%) Kuvasz (1.41%)
Xception	Golden Retriever	Golden Retriever (88.14%) Labrador Retriever (2.09%) Brittany spaniel (0.84%)

Table 5.1: Model Predictions

5.1.1 Quantitative Metrics

Discussion:

- **Consistency Across Models:** All three models predict that the image is of a Golden Retriever, showcasing consistency in their recognition capabilities.
- **Confidence Variation:** While all models identify the Golden Retriever as the primary class, ResNet50, and Xception are more confident in this prediction with scores above 88%. In contrast, GoogLeNet is slightly more uncertain with a confidence of 55.42 %.
- **Secondary Predictions:** Both GoogLeNet and ResNet50 identify Labrador Retriever and Kuvasz as the next likely breeds, albeit with varying confidence scores. Xception differs in its third prediction by suggesting the possibility of the image being a Brittany spaniel, with a low confidence score of 0.84%.

The evaluation of the three models on a single image provides a snapshot of their behaviour and recognition capability. It's encouraging to see them agreeing on the primary classification. The variations in secondary predictions and confidence levels could be attributed to architectural differences and how these models have learned features from the ImageNet dataset. For more robust conclusions, a broader dataset evaluation would be necessary, but this snapshot provides an interesting perspective on how different models perceive a specific image.

5.1.2 Visual Insights using Grad-CAM

Grad-CAM offers a glimpse into how CNNs make decisions by using heatmaps to spotlight key image areas. These visual cues help identify where the model concentrates and give us an idea about the feature selection of the model. Analysing the Grad-CAM of different models will give us insights into why one model outperforms the other.

Explanation of Grad-CAM images:

- Bright areas highlight the regions that played a significant role in the model's decision.
- Dark areas had less influence on the decision.
- The exact regions that are highlighted can give insights into what features the model has learned and whether it's focusing on the correct areas of the image for its predictions.

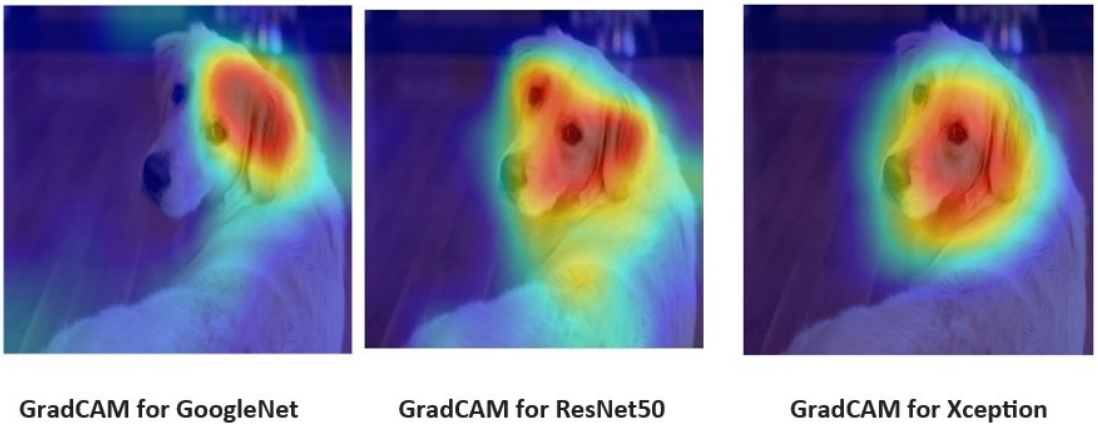

Figure 5.2: Grad-CAM images of the Networks

5.2 Untargeted Adversarial Attack Results

5.2.1 Simulating Untargeted Adversarial Attacks

We would generate an Adversarial example using the FGSM, introduced in Chapter 4, which works by utilizing the gradients of the neural network with respect to an input image, creating an adversarial image by adding a small perturbation. This perturbation is designed to be minimal but enough to deceive the model into misclassifying the image.

The key hyper parameter in FGSM is the epsilon (ϵ) value. This ϵ determines the magnitude of the perturbation added to the original image to simulate the adversarial sample. Essentially, the larger the ϵ, the more visible the perturbations, leading to a more distorted image. However, increasing ϵ also makes the adversarial nature of the image more detectable by human observers. This is demonstrated in the figure below.

Figure 5.3: Simulated Adversarial examples using GoogLeNet

5.2.2 Quantitative Metrics of Untargeted Adversarial Attacks

True Class: Golden Retriever

Model	Predicted class	Confidence Score (Top 3)
GoogLeNet	Labrador Retriever	Labrador Retriever (91.55%) Bath Towel (6.20%) Paper Towel (0.40%)
ResNet50	Basset	Basset (12.50%) Bloodhound (5.95%) Redbone (5.48%)
Xception	English Setter	English Setter (5.10%) Brittany spaniel (3.88%) Beagle (3.14%)

Table 5.2: Predictions and Confidence Scores of Different Models under Adversarial Attacks After the FGSM attack with Epsilon (ϵ) value 1

Interpretation of the Post-FGSM Attack Results:

Baseline Information:
The original, non-perturbed image was of a Golden Retriever. This is crucial, as any deviation from this in the model predictions after the attack can be attributed to the effects of the adversarial perturbations.

GoogLeNet:

- **Top Prediction:** After the FGSM attack, GoogLeNet's top prediction was Labrador Retriever with an overwhelming confidence of 91.55%. While this prediction is closely related to the original class (both are breeds of retrievers), but it's a deviation.
- **Other Predictions:** The next top predictions for GoogLeNet, "Bath Towel" and "Paper Towel", are entirely unrelated to dogs. This indicates that the FGSM perturbation not

only caused a misclassification but also pushed the model's predictions into completely unrelated categories.

ResNet50:

- **Top Prediction:** ResNet50 predicted the image as Basset, which is another breed of dog but significantly different from a Golden Retriever. The confidence, however, was much lower, at 12.50%.
- **Other Predictions:** The other two predictions, "Bloodhound" and "Redbone", are also dog breeds, suggesting that while ResNet50 was misled, it still recognized the image as a dog.

Xception:

- **Top Prediction:** Xception predicted the image as an English Setter with a confidence of 5.10 %. This confidence level is quite low, suggesting some uncertainty in the model's decision.
- **Other Predictions:** The next predictions, "Brittany Spaniel" and "Beagle", are again dog breeds. Like ResNet50, Xception still seemed to recognize the perturbed image as a dog, but not the correct breed.

All three models were misled by the adversarial attack. Not one of them correctly identified the adversarially perturbed image as a Golden Retriever. Each model responded to the FGSM attack differently. This can be attributed to the unique architectures and learning patterns of each model. Despite the adversarial attack, both ResNet50 and Xception still managed to categorize the image within the general category of dogs, even if they got the specific breed wrong. This indicates some retained semantic understanding, despite the perturbation. The confidence levels in the predictions varied considerably. GoogLeNet was highly confident in its (incorrect) prediction, while ResNet50 and Xception had relatively low confidence scores. This might indicate that GoogLeNet was more susceptible to this specific FGSM perturbation.

5.2.3 Visual Insights using Explainable AI (XAI): Comparing Pre-FGSM vs. post-FGSM.

GoogLeNet:

The Grad-CAM figure elucidates the changes brought about by the adversarial attack in the model's perception. By shifting the high-importance regions or features (depicted as the bright areas), the attack subtly alters the focus of the model. Instead of highlighting genuine attributes

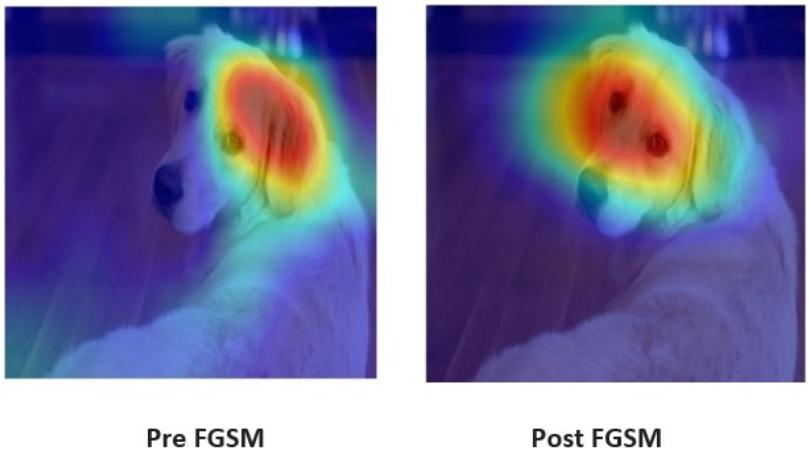

Figure 5.4: Grad-CAM of the GoogLeNet Network

of the dog in the image, the model is now drawn towards these artificially emphasized regions, often sidelining the real, prominent features. This shows that even small tweaks can make the model misinterpret the deeper meaning or context of the image.

Lime:

The LIME maps show which areas of the image are important to the classification of golden retriever. Red areas of the map have a higher importance, when these areas are removed, the score for the golden retriever class goes down. Blue areas of the map are contributing negatively to the classification. The network focuses on the dog's face and ear to make its prediction of golden retriever. This is consistent with other explainability techniques like Grad-CAM.

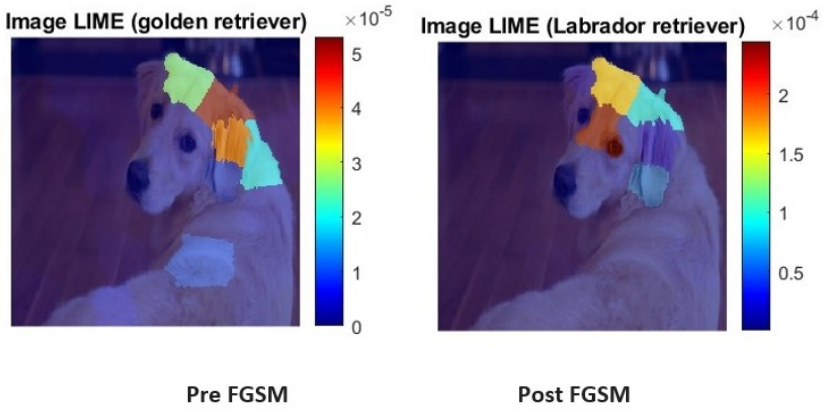

Figure 5.5: Features contributing to the model's prediction, as revealed by LIME.

For the Labrador retriever class, the network is more focused on the dog's nose, rather than the ear. While both maps highlight the dog's forehead, the network has decided that the dog's ear and neck indicate the golden retriever class, while the dog's eye indicates the Labrador re-

triever class.

Feature Selection:

Figure 5.6: Highlighting the top 5 features that determine the decision-making of the network

Feature selection involves selecting the most important features that contribute to the predictive power of a model. In this case, we can see that the top 5 features are consistent with the other Explainable Techniques (XAI). Even though it focused on the correct areas there was a misclassification we believe this can be attributed to the distribution of the training data. We will be discussing this aspect in the next chapter.

ResNet50 :

Grad-CAM:

This Grad-CAM gives us the reasoning behind the decision-making of the model and the features it focused on while making the decision. We can identify the focus areas of both are different and we will study this aspect through LIME and Feature selection.

Lime:

The maps show which areas of the image are important to the classification of golden retriever. Features that push the model's prediction away from the observed prediction are highlighted in Blue. A deeper shade of blue indicates a stronger negative influence on the predictions.

We can observe that regions that previously had a negative influence (blue) on the "golden retriever" classification now play a significant positive role in misclassifying the image in the post-FGSM scenario as a "basset".

Feature Selection:

The image shows the top 5 features the network has considered for making the decision. Here we can observe that the network gave importance to the non-robust features such as background and that might have played an important role in the misclassification.

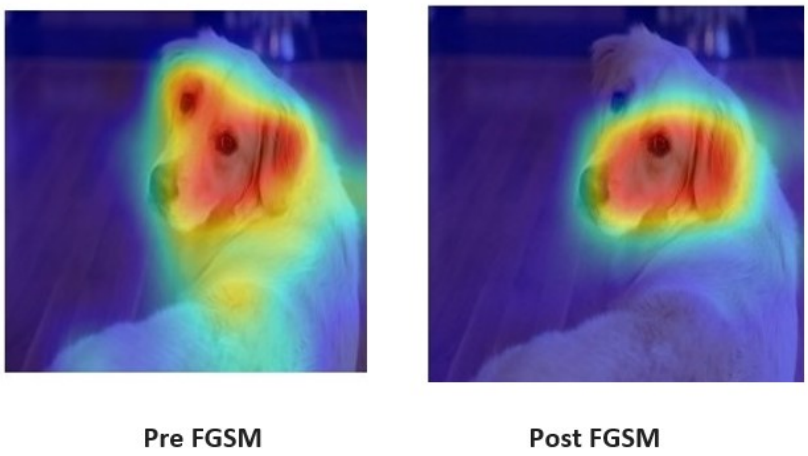

Figure 5.7: Grad-CAM of the ResNet50 Network

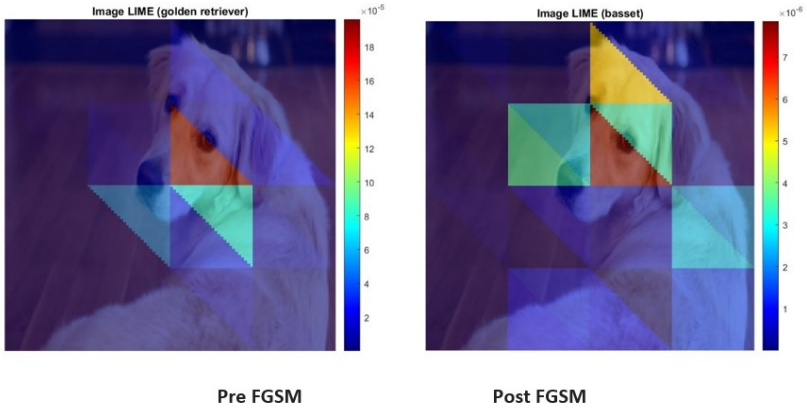

Figure 5.8: Features contributing to the ResNet50 model's prediction, as revealed by LIME

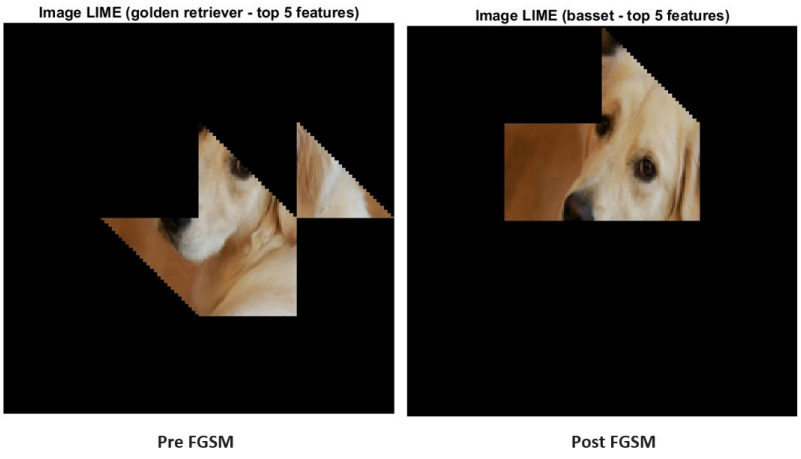

Figure 5.9: Highlighting the top 5 features that determine decision-making of the ResNet50

Xception:

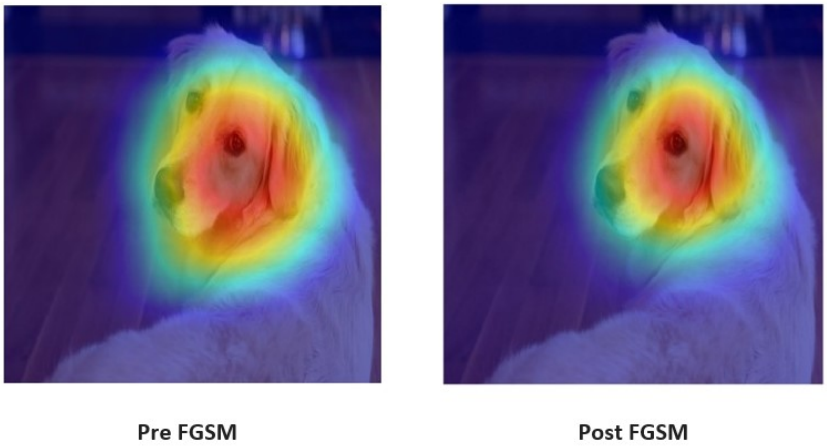

Figure 5.10: Grad-CAM of Xception network

Unlike other networks, both the original and adversarial images display more or less identical activations when viewed through Grad-CAM. This observation is further validated by the LIME and feature selection techniques.

Lime:

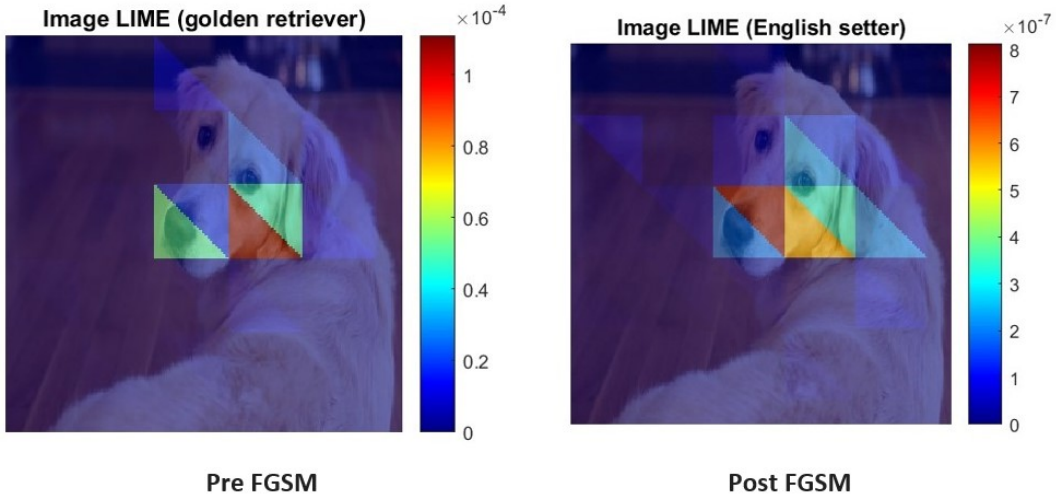

Figure 5.11: Features contributing to the Xception model's prediction, as revealed by LIME

The results of the lime are consistent with the Grad-CAM results shown above. Here we can observe that the background is highlighted in Darker Blue shade implying that it is driving the predictions away from the True class.

Feature selection:

The model's relative resistance to adversarial attacks might be attributed to its feature selection, as evidenced by its low misclassification confidence of 5.10%. Although Explainable AI Tech-

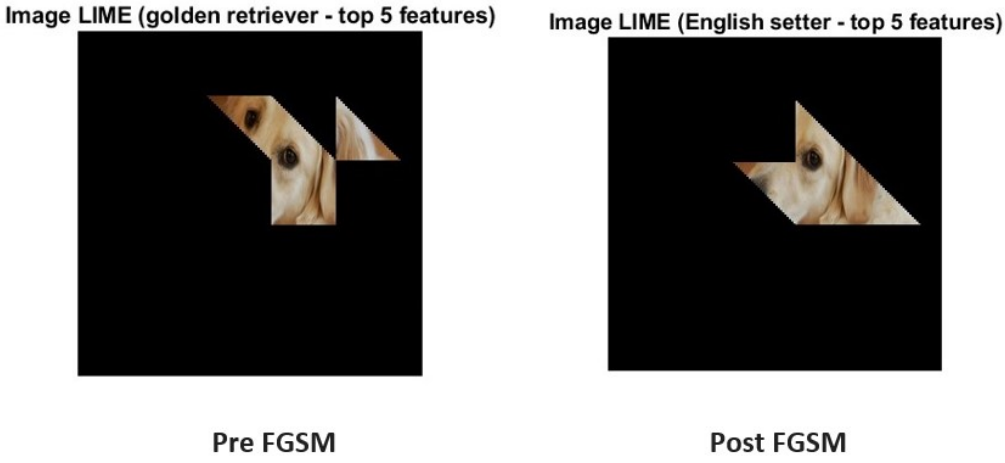

Figure 5.12: Highlighting the top 5 features that determine the decision-making of the Xception network

niques have demonstrated that the network focused on the appropriate features, but uncertainty could have potentially arisen from the network's grasp of the underlying semantic information.

5.3 Targeted Adversarial Attack Results.

5.3.1 Simulating Targeted adversarial attacks

In this experiment, the Basic Iterative method, as detailed in section 4.2, serves as our foundation for introducing adversarial perturbations. The choice of epsilon acts as a measure of the magnitude of these perturbations. A crucial component of this experiment is the variation of the alpha (α) value. By modulating α values we can study its influence on the adversarial generation process.

Once the adversarial examples are generated, it is of paramount importance to assess their efficacy. For this purpose, we track the instance where the model misclassifies an adversarial example as "Great white shark " instead of its True class "Golden Retriever". This specific instance, rather than the entire spectrum of misclassifications, provides a more focused insight into the model's vulnerability and the potency of the adversarial attack. However, merely noting the misclassification does not provide a comprehensive understanding of the underlying reasons. To delve deeper, we employ Explainable AI (XAI) techniques.

5.3.2 Quantitative Metrics of Targeted Adversarial Attacks

True Class: Golden Retriever

Targeted Class: Great White Shark

The Table 5.3 presents the effectiveness of targeted adversarial attacks, aiming to misclassify images into the "Great White Shark" class, using different pre-trained models and specific

Model	Hyper-parameter values for Targeted Attack	Confidence Score of Targeted Class
GoogLeNet	Epsilon = 3 & Alpha = 0.5	99.96%
ResNet50	Epsilon = 1 & Alpha = 0.2	3.58%
Xception	Epsilon = 1 & Alpha = 0.2	89.24%

Table 5.3: Hyper-parameter values and confidence scores for a targeted adversarial attack towards "Great White Shark"

hyper-parameters. For each model, the epsilon and alpha values, which control the attack's intensity and step size, respectively, are provided.

GoogLeNet, with epsilon set at 3 and alpha at 0.5, exhibited extreme vulnerability, misclassifying with very high confidence of 99.96%. Xception also showed a high susceptibility, slightly less than GoogLeNet, with a misclassification confidence of 89.24% using the same hyper-parameters as ResNet50. In contrast, ResNet50 displayed remarkable resilience against this targeted attack, misclassifying with a considerably lower confidence of just 3.58%. This is due to the architectural differences between the models

The interpretation outlines the variability in resilience among different models to adversarial attacks, even when targeting the same class and utilizing similar hyper-parameter settings.

5.3.3 Visual Insights using Explainable AI (XAI)

Grad-CAM:

The Grad-CAM has shown us that all three networks have activations of the non-robust fea-

Figure 5.13: Grad-CAM of Targeted Adversarial Attacks

tures such as background in this case. Grad-CAM does not explain the reason behind the Low confidence score of the Adversarial attack (3.58%) for ResNet50.As per these results, Xception should have performed better but we will gain more understanding in Feature Selection

Feature Selection:

From Feature Selection, we observe that all the networks have focused more on the Background

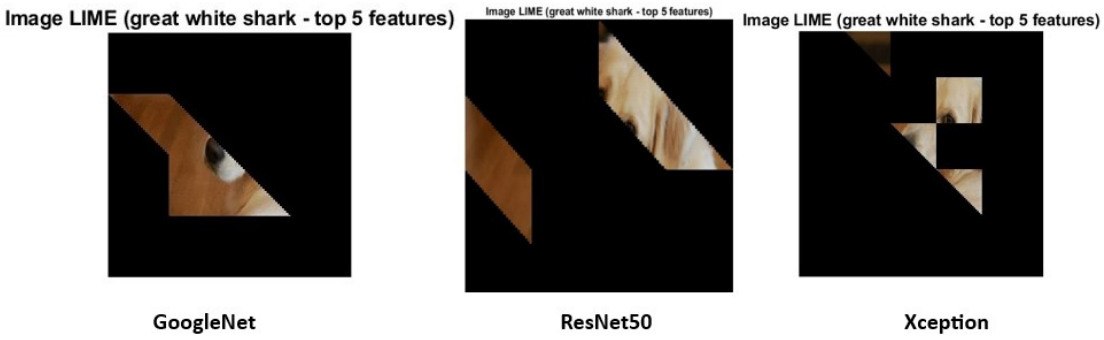

Figure 5.14: Highlighting the top 5 features that determine decision-making of the networks

and this is consistent with grad-CAM. This also gives us an explanation regarding the ResNet50 performance as we can observe that ResNet50 has focused on Robust features such as eyes and ear not just on non-robust features such as Background, Whereas Xception did not focus on Robust Features that are vital. This could be a potential reason for its better resilience against adversarial attacks compared to the other models.

5.4 Inferences

Explainable AI(XAI) gave us insights into the black box nature of the network's decision-making by visualizing them. During the exploration of untargeted adversarial attacks, especially using methods like FGSM, it's intriguing to note that while the models were deceived, they weren't misled arbitrarily. The models tended to misclassify into categories that were similar or related, in this case, different breeds of dogs. This suggests that adversarial perturbations might push the model's interpretations along existing lines of understanding, rather than inducing completely random errors.
we observed that misclassifications revolved around different breeds of dogs. This observation could provide insights into the possible data distribution during training. If the training dataset had a variety of dog breeds with potentially overlapping features, the neural networks might develop a more detailed, but also more vulnerable, understanding of these categories. Additionally, we noticed that the network's emphasis on non-robust features, such as the background observed in the LIME maps (blue color regions), played a role in misclassification

While exploring the Targeted attacks we found that non-robust features such as Background do influence the models as this is evident from the grad-CAM and Feature selection. we also observed that the strength of a model's resistance to adversarial attacks may be significantly influenced by its distribution between robust and non-robust features as seen in the ResNet50 discussed in section 5.3.3

Key Points

- Training data distribution does influence the network predictions, as a diverse dataset aids in developing a detailed understanding of various categories, It may also make the model more susceptible to adversarial attacks,
- Influence of Non-robust features such as Background,plays a crucial role in misclassification
- The model's resilience against adversarial attacks is largely determined by how it balances its attention between robust and non-robust features.

Chapter 6

Hypotheses and Evaluation

In previous chapters, we explored how advanced neural networks can be tricked by adversarial attacks. Now, we'll dig deeper. Based on insights from Chapter 5, we will discuss hypotheses about why these networks might be vulnerable.

6.1 Hypothesis 1: Dataset Distribution

"Neural networks trained on datasets with a variety of closely related classes (e.g., different dog breeds) are more susceptible to adversarial attacks due to the potential overlap of features across these classes than with datasets trained with distinctly different features ."
To Test this hypothesis:

- We will train the model using Transfer learning on a custom dataset comprising five different classes that have distinctly different features to see if there's a significant difference in vulnerability.
- After training, we will evaluate the robustness of these models against adversarial attacks, paying close attention to the confidence scores of misclassification.

6.1.1 Custom Dataset and Training using Transfer Learning

We've created a unique dataset by combining three classes from ImageNet (ImageNet , 2009) with two classes from Kaggle (Kaggle, 2013), resulting in a total of five distinct classes.

The dataset is available at this **GitHub Repository**. The dataset contains 2200 images and the images are split in 70:30 ratio for training and testing respectively.

The dataset underwent specific color preprocessing to ensure compatibility with pre-trained networks. Moreover, the input sizes were adjusted to align with the requirements of these networks. All these modifications were carried out using the **imageDatastore** and **augmentedImageDatastore** features available in MATLAB R2023a

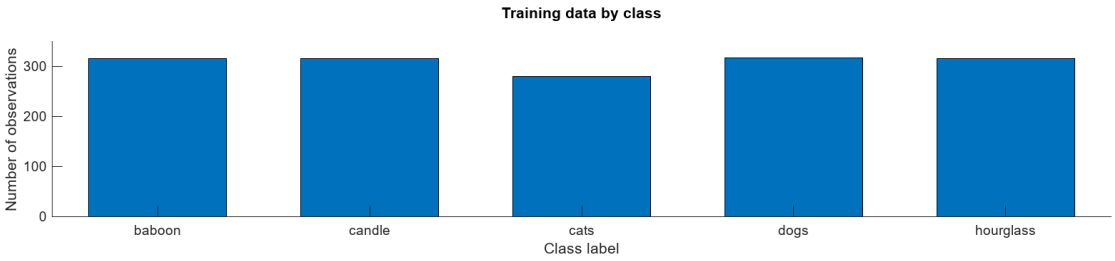

Figure 6.1: Distribution of the custom Training data

Transfer Learning

Transfer learning is a machine learning method where a model developed for a specific task is reused as the starting point for a model on a different task. In our case, we will be using the pre-trained models which are trained on large datasets such as ImageNet, and refining it for a different, typically more specific task. This approach is better because we do not need to start from scratch as it is a time-consuming process and requires computational power to do so. In this approach, we will utilize the model's initial weights and then tune them according to our custom dataset by making changes to the final layers of the network.

We have prepared the data for training and we would be using the **Deep Network Designer** in the MATLAB R2023a to train the network on our custom dataset.we aim to fine-tune a pre-trained network architecture on our custom dataset.

6.1.2 Performance of models

we would be evaluating the performance of models by using metrics such as accuracy and F1 score.

Model	Training Accuracy	Testing Accuracy	Training Times
Trained GoogLeNet	96.88%	94.24%	17 Minutes
Trained ResNet50	99.22%	94.24%	16 Minutes
Trained Xception	98.43%	97.88%	156 Minutes

Table 6.1: Performance and Execution Times of Models

Based on the model performance metrics, If speed is paramount, ResNet50 stands out. If the highest accuracy is desired and computational resources allow for longer execution times, Xception might be the preferred choice. GoogLeNet offers a balanced option between the two.

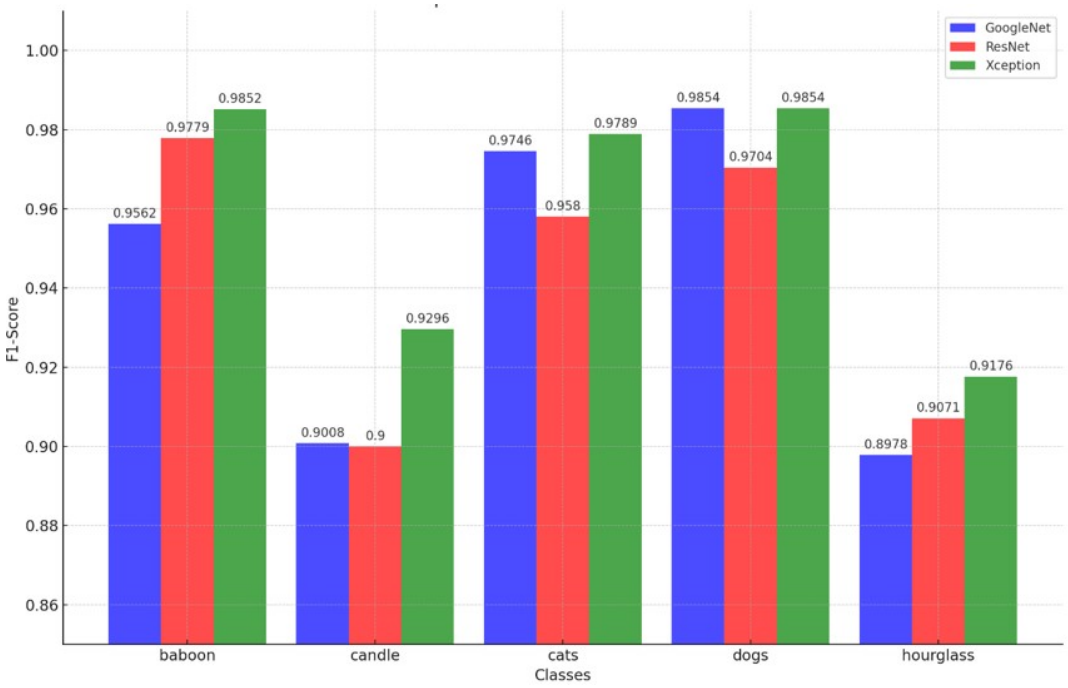

Figure 6.2: F1-Scores comparison of classes across models

F1-Scores

Data Distribution Insights:

- **Underrepresented or Challenging Classes**: Both 'candle' and 'hourglass' classes consistently exhibit lower F1-scores across all models, suggesting they might be either underrepresented or inherently challenging to classify.

- **Dataset Over Model Specificity**: The consistent trend of these two classes having lower scores across all models emphasizes that this is likely due to the dataset's quality rather than a particular model's weakness.

Robustness Insights:

- **Xception's Superiority:** Xception consistently shows higher F1-scores across most classes, implying a potentially stronger understanding of the feature space and, consequently, possible robustness against adversarial perturbations.

- **Potential Weak Points:** The notably lower F1-scores for the 'hourglass' and 'candle' classes in all the models might be a potential point of vulnerability against adversarial attacks.

6.1.3 Evaluating the Hypothesis

We have successfully concluded the training process for our models using our customized dataset through Transfer Learning. Throughout this stage, we maintained consistent hyper-parameters, including a learning rate of 0.01, a mini-batch size of 128, and the SGD optimizer.

To test our proposed theory, we will introduce adversarial examples simulated with an epsilon(ϵ) value ranging between 0 and 50 to the trained models. Following this, we will determine their class and compute the confidence scores associated with the predicted class. These outcomes will subsequently be contrasted with the results obtained from pre-trained models as presented in Chapter 5.

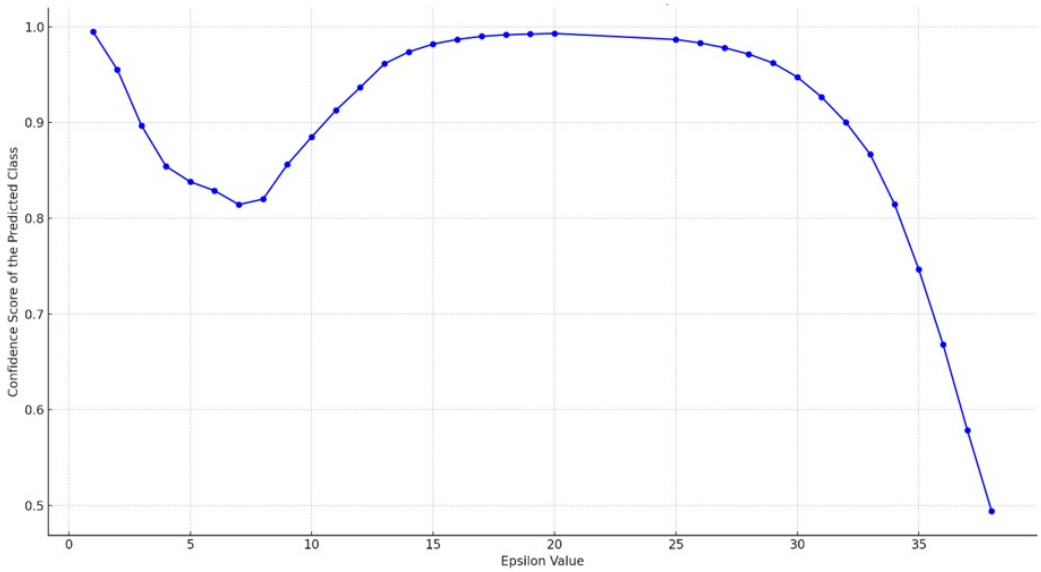

Figure 6.3: Variation in Confidence Score of Predicted Class with Increasing Adversarial Perturbation (ϵ) using Trained GoogLeNet

Key Takeaways:

- **Initial Resilience:** The model demonstrates robustness against adversarial perturbations but starts to waver as epsilon increases.
- **Vulnerability Threshold:** The sharp decline at epsilon 38 suggests a critical vulnerability threshold, beyond which the model misclassified.
- **Unexpected Stability**: The stabilization in confidence scores for intermediate epsilon values (9 to 20) is intriguing. It could be due to various factors, like the nature of the adversarial perturbation method, the underlying data, or the model's architecture.

In summary, while the model exhibits commendable resilience against minor adversarial attacks, it has clear vulnerabilities, especially at higher epsilon values. The unexpected stability in the

middle range requires further investigation to understand its root cause and potential implications.In comparison to the pre-trained model we have seen misclassification at the epsilon(ϵ) value is 1.

For both the Trained ResNet50 and Trained Xception models, simillar tests were conducted, and they demonstrated significant robustness. As an example, when subjected to an adversarial sample with an epsilon value of 100, the nwtworks predicted the true class with confidence scores of 99.99% for Trained ResNet50 and 93,25% for Trained Xception.

6.1.4 Summary

In our systematic investigation, we evaluated the adversarial resilience and overall performance of various models, such as Trained GoogLeNet, Trained ResNet50, and Trained Xception. All models were optimized using a custom dataset through Transfer Learning, with a consistent set of hyperparameters.

Data Quality and Model Robustnes:
Throughout our hypothesis testing, we sought to understand the interplay between data distribution and the performance of neural networks. A significant revelation from this exploration was the pivotal role of training data quality on the models' performance and potential robustness against adversarial attacks. Our assessment of the F1-scores highlighted that the 'candle' and 'hourglass' classes, both sourced from ImageNet, consistently underperformed. This discrepancy prompted a deeper investigation into data quality disparities between our sources. We found out that ImageNet provided lower resolution images (64x64 pixels and below 3KB in size), the Kaggle dataset, which included the better-performing 'dogs' and 'cats' classes, provided higher resolution images (300x300 pixels and approximately 30KB in size). This data quality variance might explain why high-resolution images, abundant in intricate details, help the model to understand complex features, potentially enhancing their resistance against adversarial attacks. Conversely, models trained on lower-resolution data might be more susceptible.

Comparison with Pre-Trained Models:
When compared to pre-trained models, our custom-trained iterations showcased pronounced robustness. For instance, even under intense adversarial perturbation (with epsilon values reaching 100), the Trained ResNet50 identified the correct class with a staggering 99.99% confidence, while the Trained Xception achieved a commendable 93.25%.

These findings align with our hypothesis about data distribution and overlapping features. However, a more in-depth investigation is needed before drawing definitive conclusions. we have seen glimpses of the effect of quality of data on robustness and this aspect also needs to be investigated further.

6.2 Hypothesis 2: Pre-processing of Non Robust Features

"The emphasis of a neural network on non-robust features, such as background, increases its vulnerability to adversarial attacks so if we pre=process to remove them will it improve the robustness."

To test this hypothesis:

- we will pre-process the images by removing the background and then simulating a Targeted adversarial attack as discussed in section 5.3 and checking to what extent the network is classifying correctly while varying the degree of perturbations.
- we will use the Grad-CAM to observe the focus areas of the network.

Figure 6.4: Pre-processed Base image and it's Grad-CAM

6.2.1 Evaluation

To evaluate the network's resilience to adversarial attacks, the experiment involved modifying the image by clearing the background and center-zooming it. With the same attack parameters as used in section 5.3 epsilon value 1 and alpha value 0.2, the network successfully predicted the image as a Golden retriever. However, when the epsilon value was increased to 2, while keeping alpha constant, the network still predicted the image correctly. But when the epsilon was further increased to 3, the network misclassified the image.

Figure 6.5: ResNet 50 Network predictions after perturbation (Epsilon =1) along with Grad CAM of Adversarial Image.

By observing the Grad CAM (Gradient-weighted Class Activation Mapping) images, it becomes

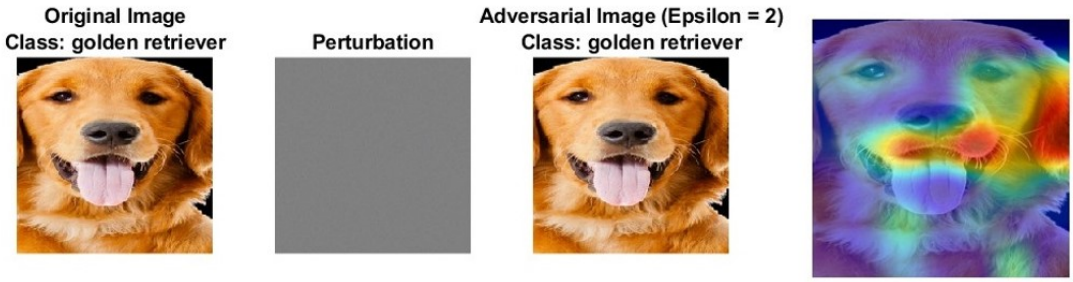

Figure 6.6: ResNet 50 Network predictions after perturbation (Epsilon =2) along with Grad CAM of Adversarial Image.

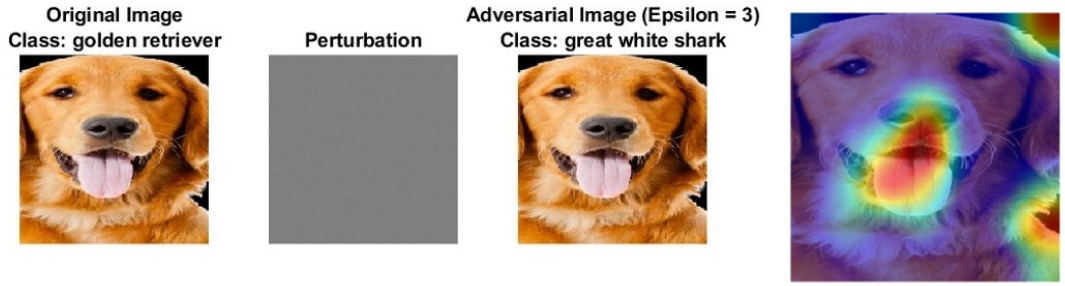

Figure 6.7: ResNet 50 Network predictions after perturbation (Epsilon =3) along with Grad CAM of Adversarial Image.

apparent that when the epsilon value was set to 3, the network appeared to overlook the crucial features of the dog and instead focused more on the background. This shift in attention and emphasis toward the background likely led to an incorrect prediction by the network.

In other words, the network's sensitivity to perturbations at a higher epsilon value resulted in a deviation from recognizing and prioritizing the relevant features of the dog. Instead, it placed excessive weight on the surrounding context, causing a misclassification.

Based on this observation, it suggests that the network's vulnerability to adversarial attacks, particularly when the epsilon value is increased, can lead to a loss of focus on important features and an increased reliance on irrelevant or misleading information, such as the background.

6.2.2 Summary

Untargeted Attacks:
The hypothesis faced challenges when it came to untargeted attacks. All three pre-trained models – GoogLeNet, ResNet50, and Xception – misclassified the images, indicating that when the model isn't provided with a specific target class to mimic, its vulnerabilities to background or non-robust features are prominent. This outcome suggests that while the preprocessing steps

might make the model more resistant to certain types of adversarial attacks, they aren't universally effective across all attack types.

Targeted Attacks:

The results were more in line with the hypothesis when it came to targeted attacks. The models were more resistant, although not uniformly. Specifically, while ResNet50 and Xception might have showcased improved resilience, GoogLeNet failed. This divergence in outcomes highlights that individual architectures might have inherent susceptibilities that cannot be easily countered merely by preprocessing.

Model-Specific Vulnerabilities:

GoogLeNet's failure, even under targeted attacks, points to a critical aspect of adversarial defense: the architecture's intrinsic characteristics. Some architectures may naturally be more robust or vulnerable to certain types of adversarial manipulations, regardless of the preprocessing techniques employed.

In summary, while the hypothesis holds merit and the preprocessing technique showcased potential in enhancing resilience against targeted adversarial attacks, it's not a one-size-fits-all solution.

6.3 Hypothesis 3 : Impact of Dropout layers

"The inclusion of dropout layers in neural networks, designed to improve model generalization by dropping connections during training, may inadvertently lead to information loss, thereby increasing the network's vulnerability to adversarial attacks."

To test this hypothesis:

- We will use the pre-trained GoogLeNet Model and change the value of dropout from 0 to 0.75 with step size of 0.25
- We will then evaluate the Top 5 Feature selection of the Network as we did in Chapter 5

6.3.1 Evaluation

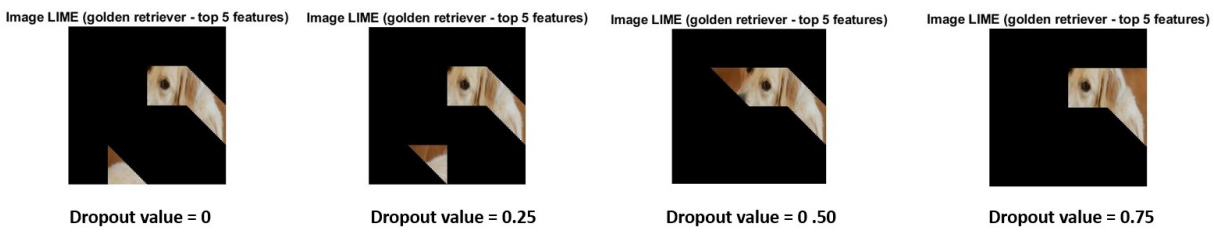

Figure 6.8: Dropout Influence on GoogLeNet's Feature Selection

we have been trying to explore the resilience of neural networks against adversarial attacks and see how certain model parameters, like dropout rates, impact their robustness.

Dropout & Feature Selection:

Dropout is a regularization technique used in neural networks. It involves randomly ”dropping out” (i.e., setting to zero) a number of output features of the layer during training. The idea is that this prevents overfitting by ensuring that no single feature heavily dictates the output.

In the context of our discussion, we experimented with different dropout rates in the GoogLeNet model to observe how it affects the features the model considers most important (feature selection). Changing the dropout rate indeed affected which features of an image the model gave more importance to.

Impact of Dropout on Robustness:

Our experiments suggested that by adjusting the dropout rate, we also inadvertently adjusted which features of an image the network prioritizes. This means that the model's resistance or susceptibility to adversarial attacks can vary based on the dropout rate. A certain dropout rate might make the model more robust against one kind of adversarial attack but make it more vulnerable to another.

The challenge here is to find an optimal dropout rate that offers a balance between model generalization (its ability to perform well on unseen data) and robustness (its resistance to adversarial attacks). Increasing the dropout rate too much might increase robustness at the cost of generalization, and vice versa.

6.3.2 Conclusion & Future Direction:

Understanding how dropout rates affect feature importance in neural networks is crucial when designing models to be both efficient and resistant to adversarial attacks. Future research might focus on fine-tuning dropout values to defend against specific adversarial perturbations.

Chapter 7

Discussion

In the vast realm of neural networks, understanding the underlying decision-making processes remains a stern challenge, often compared to a ”black box”. In this study, we aimed to explore these using Explainable AI(XAI) and then with this knowledge, we aimed to investigate Adversarial attacks and their impact on Feature selection. we will revisit the objectives and delve into a discussion on them.

Objective 1: ” To gain an understanding of adversarial attacks and their impact on neural networks ”

We have extensively reviewed scholarly articles and research papers to gain an understanding of adversarial attacks. We have seen the existence of adversarial attacks and thereon the real-world implications and how to simulate them.

Objective 2: ”To investigate how neural networks are fooled by adversarial attacks by using Explainable AI (XAI)”

Building on our understanding of adversarial attacks, we embarked on a detailed exploration of various neural network architectures. By analyzing models such as GoogleNet, ResNet50, and Xception, we sought to understand the unique attributes and potential vulnerabilities of each. Recognizing that each network, with its distinct design principles, offers varied insights into image classification, our investigation aimed to explain the inherent decision-making processes. This understanding is vital, as only by grasping the core functions of these networks can we truly understand how and where adversarial perturbations might exploit them. we employed Explainable AI(XAI) to understand and Interpret the Adversarial attacks.

Through Grad-CAM, we could visually pinpoint the regions in the image that influenced the network’s decisions, shedding light on the areas most susceptible to adversarial perturbations. LIME further complemented this by providing localized interpretations, allowing us to understand model predictions on a per-instance basis. Additionally, we explored Feature Selec-

tion and highlighted the significance of various image attributes in the classification process and how adversarial attacks can manipulate these to deceive the network.

This phase of work was instrumental in bridging the gap between raw neural operations and interpretable outputs. By understanding how networks react and modify their decisions under adversarial influence, we came up with the hypotheses.

Objective 3: "To formulate a hypothesis or propose a method to enhance the robustness of neural net- works against adversarial attacks"

Based on your work in objective 2 we have formulated 3 hypotheses and evaluated them in this work. The key insights gained from this hypotheses evaluation were :

- **Impact of training data distribution:** It became evident from our results that models trained on datasets characterized by diverse distributions and non-overlapping features exhibited enhanced robustness against adversarial attacks. This finding explains the need of curating a well-distributed dataset, free from overlapping features, to fortify models against adversarial threats.
- **Impact of dropout layers on the feature selection** These layers, typically employed to prevent overfitting, inadvertently rendered the network more vulnerable to adversarial perturbations. This observation emphasizes the delicate balance needed between enhancing generalization and ensuring adversarial robustness.

Limitations

One of the primary limitation of this work is that we did not delve deeply into the explicit architectural differences between the various neural networks. While we analyzed models such as GoogleNet, ResNet50, and Xception, a more detailed comparison of their architectures and their implications on adversarial robustness remains unexplored in this work as it is beyond the scope. Understanding these differences could provide more insights into why certain architectures might be more vulnerable or resilient to adversarial attacks than others.

we are also constrained by computational limitations. While we aimed to train our models on a large and diverse dataset to capture a more comprehensive understanding of adversarial robustness (Hypothesis 2), the lack of sufficient computational power restricted us.

Future Scope

One intriguing avenue for further exploration lies in the realm of component-wise training of neural networks. Rather than training a model on entire images, what if we were to break down the input into its constituent parts? For example, if we consider a dog can we train a network

separately on the eyes, ears, and nose and then combine these individual features to train as a whole? Furthermore, by establishing relations or connections between the individual features during training, the network can learn to incorporate contextual information and dependencies between them. This can enable the network to make more informed and accurate predictions by considering the interplay and relationships between different parts of the input.

Such an approach could potentially revolutionize the way neural networks perceive and understand inputs. By first acquainting the model with individual features, and subsequently integrating this knowledge, the network might be equipped with a richer contextual understanding. This context, which arises from understanding inter-feature relationships, might hold the key to more accurate and holistic image interpretations.

Beyond just accuracy, there's also the aspect of robustness. If adversarial attacks typically exploit minute perturbations in images, a model grounded in the interplay of various features might be inherently more resistant. Rather than being misled by alterations to a single feature, such a model would weigh in the collective representation of all features, potentially nullifying the impact of the adversarial input.

While this concept sounds promising, its practical implementation could be challenging. How do we effectively break down images into constituent features? How do we ensure the model understands the inter-relationships? And, most importantly, how do we integrate these individual learnings?

A potential answer to these challenges might lie in Multi-Instance Learning (MIL). In MIL, bags of instances are presented to the model, with a bag label indicating if at least one instance in the bag belongs to the positive class. Leveraging MIL for component-wise training might offer a structured framework to train on individual features and subsequently integrate their learnings.

In conclusion, the future scope of this research beckons a deep dive into component-wise training, potentially harnessing the power of techniques like MIL. Such investigations could redefine our understanding of neural network robustness and offer innovative defense mechanisms against adversarial threats.

Bibliography

Adrien Bennetot, Ivan Donadello, Ayoub El Qadi, Mauro Dragoni, Thomas Frossard, Benedikt Wagner, Anna Saranti, Silvia Tulli, Maria Trocan, Raja Chatila, Andreas Holzinger, Artur d'Avila Garcez, Natalia Díaz-Rodríguez. 2022. A Practical Guide on Explainable AI Techniques applied on Biomedical use case applications. *arXiv:2111.14260*. Available from:https://arxiv.org/abs/2111.14260

Alireza Sarraf Shirazi & Ian Frigaard. 2021. SlurryNet: Predicting Critical Velocities and Frictional Pressure Drops in Oilfield Suspension Flows . *Energies*. **14**(5), p.1263.

Andrej K. 2015. Neural networks part 1: Setting up the architecture, Notes for CS231n convolutional neural networks for visual recognition. *Stanford University.*

B, Gülmez. 2023. A novel deep neural network model based Xception and genetic algorithm for detection of COVID-19 from X-ray images. *Annals of Operations Research,* **328**, pp. 617–641. Available from:https://doi.org/10.1007/s10479-022-05151-y.

Brendel Wieland , Rauber Jonas , Bethge Matthias. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models, *Proceedings of the International Conference on Learning Representations (ICLR)*. Available from:https://arxiv.org/abs/1712.04248

Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, Prateek Mittal. 2019. DARTS: Deceiving Autonomous Cars with Toxic Signs.[online]. *Adversarial Machine Learning @ Princeton*, Available from:https://adversarial-learning.princeton.edu/darts/

Chris V. Nicholson. A Beginner's Guide to Neural Networks and Deep Learning. 2023. [online], [Accessed 07 August 2023] Available from:https://wiki.pathmind.com/neural-network.

Chollet, François. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1251–1258. Available from:https://doi.org/10.1109/CVPR.2017.195

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1–9.

Goodfellow, Ian and Shlens, Jonathon and Szegedy, Christian. 2015. Explaining and Harnessing Adversarial Examples, *Proceedings of the International Conference on Learning Representations (ICLR)*, Available from:https://arxiv.org/abs/1412.6572

He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian. 2015. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770–778. Available from:https://doi.org/10.1109/CVPR.2015.7298594

H. Lohninger.1999. Teach/Me Data Analysis. *Springer-Verlag, Berlin, New York, Tokyo,*.ISBN 3-540-14743-8.

I. Goodfellow, Y. Bengio, and A. Courville. 2016. Deep Learning. *MIT press*.

Ilyas, Andrew and Santurkar, Shibani and Tsipras, Dimitris and Engstrom, Logan and Tran, Brandon and Madry, Aleksander. 2019. Adversarial Examples Are Not Bugs, They Are Features, *arXiv preprint arXiv:1905.02175*, Available from:https://arxiv.org/abs/1905.02175

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. *CVPR*.

Moustapha Cisse, Yossi Adi, Natalia Neverova, Joseph Keshet. 2017. Houdini: Fooling Deep Structured Prediction Models. *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

Nicholas Carlini, David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. *Proceedings of the IEEE Symposium on Security and Privacy (SP)*.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, Ananthram Swami. 2016. The Limitations of Deep Learning in Adversarial Settings. *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*.

Krystian Safjan, LIME - Understanding How This Method for Explainable AI Works. 2023. *Krystian's Safjan Blog*. [Online.] [Accessed 21 August 2023.] Available from:https://safjan.com/how-the-lime-method-for-explainable-ai-works/

Kurakin, Alexey and Goodfellow, Ian and Bengio, Samy. 2016. Adversarial Examples in The Physical World. *arXiv preprint arXiv:1607.02533*. https://arxiv.org/abs/1607.02533

Pingel, Johanna. 2023. What is Explainable AI?. *MathWorks Blog*.[Online.] [Accessed 15 July 2023.] Available from:https://blogs.mathworks.com/deep-learning/2022/12/30/what-is-explainable-ai

Ribeiro, Marco Tulio and Singh, Sameer and Guestrin, Carlos. 2016. Why Should I Trust You? Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. pp. 1135–1144. Available from:https://doi.org/10.1145/2939672.2939778

Selvaraju, Ramprasaath R. and Cogswell, Michael and Das, Abhishek and Vedantam, Ramprasaath R. and Parikh, Devi and Batra, Dhruv. 2019. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. pp. 618–626. Available from:https://doi.org/10.1109/ICCV.2017.74

Siddharth Sharma, Simone Sharma & Anidhya Athaiya. 2020. Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*. **4**(12), pp. 310–316, ISSN 2455-2143.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard. 2015. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Simonyan, Karen and Zisserman, Andrew. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*. Available from:https://arxiv.org/abs/1409.1556

Stanford's CS231n course, *Convolutional Neural Networks for Visual Recognition: Softmax*. [Online] [Accessed on 15 July 2013]. Available: http://cs231n.github.io/linear-classify/#softmax

Su, Jiawei and Vargas, Danilo and Sakurai, Kouichi. 2019. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation*. **23**(5), pp. 828–841. Available from:https://doi.org/10.1109/TEVC.2019.2897123

Szegedy, Christian and Zaremba, Wojciech and Sutskever, Ilya and Bruna, Joan and Erhan, Dumitru and Goodfellow, Ian and Fergus, Rob. 2014. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*. Available from:https://arxiv.org/abs/1312.6199

Will Cukierski. 2013. Dogs vs. Cats. *Kaggle*. [Online] [Accessed on 5 August 2013]. Available:https://kaggle.com/competitions/dogs-vs-cats