Vulnerabilities and Robustness in Computer Vision

by

Konstantakopoulos Dimitris MTN2013

Submitted
in partial fulfilment of the requirements for the degree of

Master of Artificial Intelligence

at the

UNIVERSITY OF PIRAEUS

June 2024

MSc “Artificial Intelligence”

Month 7, 2024

Certified by. . . . . .

Stasinos Konstantopoulos
Research Associate
Thesis Supervisor

Certified by. . . . . .

Theodore Giannakopoulos
Researcher
Member of Examination Committee

Certified by. . . . . .

Antonios Troumpoukis
Research Associate
Member of Examination Committee

Vulnerabilities and robustness in Computer Vision

By

Konstantakopoulos Dimitris

Submitted to the MSc “Artificial Intelligence” on September, 2024, in partial fulfillment of the
requirements for the MSc degree

Abstract

The dissertation intends to examine the reliability and robustness of the most recent computer vision models in environments different from those they have been trained on. The study will focus on the performance of the models on idiosyncratic datasets and in environments with malicious users. Specifically, the research phases will include the creation of multiple state-of-the-art computer vision models with different architectures, and after the verification of their performance on common datasets, we will proceed to test them on idiosyncratic datasets, such as ObjectNet, while also examining their resilience to black and white box adversarial attacks . Based on the results of these tests, we will evaluate the effectiveness, reliability, and robustness of these computer vision models. Additionally, we will examine the transferability of some of these attacks among different model architectures. This approach will allow the identification of potential weaknesses in the models' ability to generalize their knowledge to uncontrolled and adversarial environments and open the discussion for possible defenses and mitigation measures for these weaknesses, as well as the capabilities of each architecture. Variants in neural networks and attacks will be selected based on the specific needs of the research and the continuous updates in the field.

Supervisor:
Stasinos Konstantopoulos,
Academic Position:
Researcher Associate at NCSR
'Demokritos

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my thesis supervisor, Professor Stasinos Konstantopoulos, whose guidance, support, and insightful feedback have been invaluable throughout the development of this thesis. His expertise and encouragement have greatly enriched my academic journey and have been crucial in helping me navigate the complex and fascinating field computer vision robustness.

I would also like to extend my heartfelt thanks to all the instructors in the MSc Artificial Intelligence program. Their dedication, knowledge, and passion for the subject have profoundly shaped my understanding and inspired me to push the boundaries on our field. The lessons I have learned during my time in this program will stay with me throughout my career.

To my friends, thank you for your unwavering support and for being my sounding board during the countless discussions we had about our shared interests and challenges. Your camaraderie and encouragement have made this journey all the more rewarding.

Finally, I owe my deepest appreciation to my parents. Your love, patience, and belief in my abilities have been my greatest source of strength. Without your constant support, this thesis would not have been possible. Thank you for always standing by me, especially during the most challenging moments.

This thesis is as much yours as it is mine.

Table of Contents

LIST OF FIGURES

Abbrevations List

English Abbreviation	Desciption
VIT	Vision transformer
CVT	Convolutional Transfomer
AI	Artificial Intelligence
NN	Neural network
GOFAI	Good Old fashioned Artificial Intelligence
ML	Machine Learning
FGSM	Fasr Gradient Sign Method
LSTM	Long-Short Term Memory
CNN	Convolutional neural Network
RGB	Red Green Blue
ReLU	Rectified Linear Unit
BERT	Bidirectional Encoder Representations form transfomer
ROC-AUC	Receiver Operating Characteristic Area Under the Curve

1 Introduction

1.1 Artificial Intelligence

Artificial Intelligence (AI) stands as one of the most transformative technologies of our current century, revolutionizing industries, augmenting human capabilities, and reshaping the way we interact with computers. This journey begins by understanding the core principles of AI and its profound impact on the current era, often referred to as the Fourth Industrial Revolution. AI is a sub-field of computer science that encompasses a vast array of techniques enabling machines to simulate human cognitive abilities by making decisions, learning, reasoning, and perceiving the world around us.

AI boasts a diverse landscape of techniques, with two fundamental approaches dominating the field: symbolic AI and non-symbolic AI. Symbolic AI, also known as Good Old-Fashioned AI (GOFAI), emphasizes representing knowledge and reasoning explicitly using symbols and logic rules. Key subfields include knowledge representation, reasoning, logic programming, and expert systems. Non-symbolic AI adopts a more data-driven approach. Instead of explicit rules, these systems learn from large amounts of data and identify patterns to perform tasks or make predictions. Prominent subfields include machine learning algorithms, deep learning, reinforcement learning, natural language processing, and computer vision.

The seeds of AI can be traced back to early philosophical inquiries into the nature of intelligence and the possibility of creating machines that could think. One of the first references can be found in ancient Greece, where the philosopher Aristotle described the concept of automata as self-operating machines. However, the formal birth of AI is often attributed to the seminal work of Alan Turing in 1950. His paper, "Computing Machinery and Intelligence," introduced the Turing Test, a proposed standard for determining if a machine can exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human. The subsequent decades witnessed significant advancements in fields like computer science, mathematics, and logic, paving the groundwork for the development of more sophisticated AI techniques, especially in machine learning, where its applications are starting to accommodate our daily lives.

The Machine learning Renaissance

The groundwork for Machine Learning (ML) was laid in the early days of computer science with the development of algorithms like linear regression, decision trees, and nearest neighbors but early computers lacked the processing power and data storage capacity needed for training complex ML models. The field of AI was dominated by symbolic AI approaches during this period, with research focusing on knowledge representation and reasoning. During the 1970s and 1980s, the focus shifted to expert systems, which aimed to emulate human expertise in specific domains. While these systems achieved some success, they were limited by their reliance on handcrafted rules and knowledge bases, highlighting the need for more flexible and adaptive approaches.

The seed of change started in the early 1990s, where advancements in hardware, particularly with the rise of powerful workstations and early GPUs, allowed more robust training of complex models using the explosion of digital data as the necessary fuel for data-driven ML approaches. Developments in algorithms like the Support Vector Machine (SVM) and the resurgence of neural networks (especially recurrent neural networks like LSTMs) opened doors for tackling more real-world problems. This period also witnessed the rise of backpropagation and the widespread application of neural networks to tasks such as pattern recognition, speech recognition, and financial modeling.

The proliferation of digital data in the 2000s provided fertile ground for machine learning to flourish. The advent of big data technologies enabled the collection, storage, and analysis of vast amounts of information, empowering machine learning algorithms to extract valuable insights and make more accurate predictions.

The 2010s marked the onset of the deep learning revolution, driven by breakthroughs in neural network architectures, optimization techniques, and parallel computing. Deep learning models, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), achieved unprecedented performance in image recognition, natural language processing, and other domains, catalyzing the rapid adoption of Ai technologies across numerous industries. These factors, combined with growing public and private investment in AI research, fueled the rapid advancement of machine learning following breakthroughs in areas like computer vision, natural language processing, speech recognition, and reinforcement learning. As we enter the present decade and beyond, the machine learning landscape continues to evolve at a rapid pace. Advancements in areas such as reinforcement learning, generative models like transformers, and meta-learning promise to further expand the capabilities of AI systems, unlocking new frontiers in automation, personalization, and decision-making.

Computer vision

One of the most transformative applications of machine learning and the core AI field of this thesis is computer vision. With the advent of deep learning, computer vision systems have achieved remarkable progress in tasks such as object detection, image classification, and image segmentation. The ability of neural networks to automatically learn hierarchical features from raw data has enabled unprecedented levels of accuracy and robustness in visual recognition tasks, paving the way for applications ranging from autonomous vehicles to medical imaging.

At its core, computer vision aims to enable machines to interpret and understand the visual world, much like humans do. By analyzing digital images or videos, computer vision systems can extract meaningful information, identify objects and scenes, and make decisions based on visual inputs. This capability has wide-ranging implications across various domains, from healthcare to entertainment, transportation to security.

Computer vision encompasses a broad spectrum of tasks and techniques, each addressing different aspects of visual perception and understanding. Some key fields within computer vision may include:

Image Classification and Object Recognition: This field which our thesis is focused is used on identifying objects or patterns within images and assigning them to predefined categories. Deep learning models, particularly convolutional neural networks (CNNs), have revolutionized image classification by automatically learning hierarchical features from raw pixel data, enabling accurate recognition of objects in complex scenes.

Object Detection and Localization: Object detection involves not only recognizing objects within an image but also locating their precise positions. This field is crucial for applications such as autonomous driving, where vehicles must accurately detect and track pedestrians, vehicles, and other obstacles in real-time.

Image Segmentation: Image segmentation involves partitioning an image into semantically meaningful regions, allowing for finer-grained analysis and understanding of visual content. Segmentation techniques are widely used in medical imaging, remote sensing, and video surveillance, among other domains.

Pose Estimation and Gesture Recognition: Pose estimation aims to infer the spatial pose or configuration of objects or humans within an image or video. Gesture recognition builds upon pose estimation to interpret human gestures and movements, enabling natural and intuitive human-computer interaction in applications such as augmented reality and sign language recognition.

Scene Understanding and Visual Reasoning: Scene understanding involves higher-level reasoning about the relationships and semantics of objects within a scene. This field encompasses tasks such as scene classification, semantic segmentation, and visual question answering, pushing the boundaries of machine intelligence in comprehending complex visual environments.

The field of computer vision continues to evolve rapidly, driven by advances in machine learning, sensor technology, and computational power and its impact on our society will only grow unlocking new opportunities for innovation and human-machine interaction in the visual realm making its robustness a core principle to its design . Robustness is defined as the ability to maintain performance and reliability under diverse and challenging conditions. Achieving trust, safety, and effectiveness in applications ranging from autonomous vehicles and medical diagnosis to financial trading and natural language processing is a critical concern ,posing also as its most significant challenge as these delicate systems are susceptible to various vulnerabilities and design flaws especially when they are trained with real world data which is often incomplete.

Addressing the challenges of robustness and vulnerabilities in AI systems requires a multi-disciplinary approach, encompassing advances in machine learning, data ethics, cybersecurity, and human-computer interaction. By developing robust, trustworthy, and accountable AI systems, we can harness the transformative potential of AI technologies while mitigating risks and ensuring their responsible and ethical use in society.

Ensuring their robustness and reliability of such AI technologies becomes paramount. Challenges such as adversarial attacks, domain shifts, and data biases can undermine the performance of these systems in unexpected ways, highlighting the need for robust and generalizable solutions. Addressing these challenges requires interdisciplinary efforts aimed at developing algorithms and methodologies that are not only accurate in their domain but also resilient to diverse environmental conditions and adversarial influences.

1.2 Delving into robustness

Robustness as per se refers to the ability of a system to maintain its performance under different conditions, including variations in input data, environmental factors, and adversarial attacks. Conversely, vulnerabilities highlight weaknesses within these systems that can be exploited to undermine their performance or integrity. Understanding the interplay between robustness and vulnerabilities in computer vision is imperative for ensuring the reliability, safety, and fairness of AI technologies in real-world scenarios.

This thesis delves into the scrutinization of robustness and vulnerabilities in computer vision, aiming to elucidate the underlying mechanisms and conditions, explore existing methodologies, and come to a coclusion the resilience of various architectures . By dissecting the challenges posed by various sources of uncertainty, such as occlusions, lighting variations, and adversarial perturbations, this research endeavors to foster a deeper comprehension of the strengths and limitations inherent in contemporary computer vision algorithms. The journey through this thesis begins with a comprehensive review of the literature, surveying the landscape of research efforts dedicated to enhancing the robustness of computer vision systems. Drawing insights from diverse perspectives, including machine learning and image processing , we will try to lay the foundation for a holistic understanding of the factors influencing the performance and reliability of AI-driven visual perception.

Subsequently, we investigate the vulnerabilities that lurk beneath the surface of seemingly robust computer vision models. Through empirical analysis and theoretical frameworks, we uncover the susceptibilities to adversarial attacks, data biases, and domain shifts that undermine the trustworthiness of AI systems in real-world deployments. By identifying these vulnerabilities, we pave the way for developing mitigation strategies and countermeasures to fortify the resilience of computer vision algorithms against emerging threats.

Furthermore, this thesis endeavors to bridge the gap between theoretical insights and practical applications by presenting experimental validations and case studies across diverse use cases. Through rigorous experimentation and validation on benchmark datasets and real-world scenarios, we aim to demonstrate the efficacy of proposed methodologies in bolstering the robustness and reliability of computer vision systems in mission-critical applications.

In conclusion, this thesis serves as a testament to the indispensable role of robustness and vulnerability analysis in shaping the future trajectory of computer vision research and applications. By embracing the challenges posed by real world uncertainty and adversarial environments, we embark on a journey towards building AI systems that are not only intelligent but also resilient, trustworthy, and ethically sound.

THE MEANING OF ROBUNSTNESS

As the field of computer vision is rapidly expanding with new models and their variants, it is becoming increasingly integrated into our everyday lives. Vision transformer models, although a relatively new architecture, are here to stay and over time, their ability to produce high quality results continues to grow. The current market, constrained by computational and information resource limitations, shifts its focus to the efficiency of such models along with their accuracy. However, considering these as the only key metrics is insufficient for making them deployable in most real world scenarios. Characteristics such as scalability, explainability and especially their robustness must be taken more seriously before they intergrate into the worldwide market.

In computer vision, where models are tasked with interpreting and making decisions based on visual data, robustness becomes crucial for several key reasons:

Real-world data is messy and unpredictable: Unlike the controlled environments used during training, real-world images can be affected by various factors such as:

- **Lighting variations:** Images can be captured in different lighting conditions, affecting object appearance and model predictions.
- **Noise and distortions:** Images might contain noise, blur, compression artifacts, or other distortions that can confuse the model.
- **Occlusions and partial views:** Objects may be partially hidden or occluded by other objects, making recognition difficult.
- **Adversarial attacks:** Malicious actors can intentionally manipulate images to mislead the model, causing misclassification or security risks.

A robust model needs to handle these variations and challenges without significantly dropping its performance.

Safety and reliability are critical: Computer vision systems are increasingly used in safety-critical applications like autonomous vehicles, medical diagnosis, and security systems. In these scenarios, even small errors in predictions can have severe consequences. Robustness ensures the model functions reliably and doesn't make critical mistakes due to unexpected inputs.

Generalizability and trust: A model trained on one specific dataset might not perform well on unseen data with different characteristics. Robustness ensures the model can generalize to new situations and maintain its performance across diverse scenarios, building trust in its predictions.

Ethical considerations: Biases and unfairness present in training data can be amplified by non-robust models, leading to discriminatory outcomes. A robust model should be able to resist these biases and make fair decisions even in challenging scenarios.

Evolving environments: Real-world environments and scenarios are constantly changing. A robust model can adapt to these changes without requiring constant retraining, ensuring its long-term usefulness and value.

Therefore, robustness is not just a desirable feature but a necessity for computer vision models to operate safely, reliably, and ethically in the real world. By prioritizing robustness, we can develop computer vision models that are not only accurate but also capable of handling the complexities and uncertainties of the real world, leading to safer, more reliable, and trustworthy applications.

2 Computer vision Robustness

2.1 Image classication Domain

Conducting experiments and drawing conclusions on robustness in computer vision is not a simple task. It requires careful consideration of various factors, ranging from the selection of the specific problem domain to the choice of models, their training parameters, and their evaluation methodologies.Given the complexity of the task, we have focused our experimental efforts on the problem of image classification. This choice offers several advantages for robustness testing.

Image classification is a fundamental task in computer vision with wide-ranging real-world applications, making it a pertinent area for investigation. The inherent complexity and variability of image data and specific datasets present ample opportunities to assess the robustness of models under diverse conditions . Unlike some other domains, such as text or numerical data, images are highly interpretable to humans. making it easier to understand and analyze the behavior of image classification models, both in normal conditions and malicious ones.

Furthermore the Image classification domain is one of the most well-researched domains in computer vision, providing us the opportunity to choose from a wide variety of models with credible training sources, benchmarking, and results. Additionally, image classification models are known to be susceptible to various types of attacks, such as adversarial attacks, where imperceptible perturbations are added to input images to mislead the model's predictions. These vulnerabilities highlight the need for robustness testing to identify and mitigate potential weaknesses, allowing us to apply well-tested techniques on extensively studied datasets.

To start the robustness experiments, it is necessary first to define the domain we are testing. Image classification lies at the heart of computer vision, serving as a fundamental task where machines categorize and assign labels to images based on their visual content. This process enables computers to distinguish between different objects, scenes, or concepts depicted in images, laying the groundwork for numerous applications such as content-based image retrieval, medical imaging diagnosis, and autonomous navigation.

Machines do not see images as humans do. The input to an image classification model is typically a matrix of pixel values representing the image. Each pixel value corresponds to the intensity or color of the corresponding pixel in the image. For grayscale images, this matrix is 2-dimensional, while for color images, it is usually a 3-dimensional matrix representing the red, green, and blue (RGB) color channels.

The domain techniques can be divided in four types:

- **Supervised Learning:** Which the the computer vision model is trained to identify visual patterns from a set of labeled image datasets. During the process, the model learns to associate images with the labels assigned to them.

- **Unsupervised Learning:** Which the the computer vision model is trained without using labeled training data. Instead, the model freely analyzes the image datasets to form useful deductions.

- **Semi-supervised Learning:** Combines both supervised and unsupervised methods for the training process .

Deep learning models train the same way as basic machine learning models do. They learn to associate images patterns through supervised, unsupervised, or semi-supervised training. What differs is the deep learning model's robust ability to extract, analyze and understand complex relationships and representations of image data in a much larger magnitude.

This thesis focuses on testing the robustness and identifying vulnerabilities in deep learning image classification models. For these models to effectively recognize images, they require not only appropriate selection but also training with annotated data to establish necessary connections. Additionally, they need to be evaluated using specific techniques to estimate their capabilities.

Traditionally, convolutional neural networks (CNNs) have been predominantly used in image classification tasks. However, recent advancements have introduced transformers and their hybrids with CNNs, signaling a paradigm shift in the field.

2.2 Computer Vision State of the art models and their selection.

To effectively test the robustness of image classification models, several considerations must be made. The selected model architectures should represent a variety of those used in today's image recognition applications to ensure our research is relevant to the field. Additionally, the training processes for these models should be well-documented so that our results can be accurate and comparable. The training datasets must be accessible, with reliable annotations and classes that are also present in other datasets.

Robustness-related experiments need to examine many factors that could impact the model's efficacy. This thesis is dedicated to testing the robustness of specific model architectures by evaluating their generalization and biases on unseen, challenging datasets, as well as in environments involving generative adversarial and genetic attacks. We aim to identify potential vulnerabilities in their designs. Given the increasing prominence of transformer architectures in computer vision, it is crucial to determine if this trend aligns with their robustness.

The model architectures were not chosen arbitrarily; multiple considerations were taken into account. Firstly, the availability of the models was essential. Not all state-of-the-art models are publicly accessible. Even when available, we required solid baseline metrics, specifically on the ImageNet dataset as it is the industry standard, detailed information about their architecture, and the necessary tools for experimentation.

We selected models with well-known and tested architectures, available in familiar libraries, and equipped with the tools necessary for our experiments. These models were trained by official, trusted sources that could provide the necessary information. The selected libraries include the Transformer library and Keras, with models from official vendors such as Google, Keras, and Microsoft.

Furthermore, the effectiveness of these models needed to be verified and tested by other official sources beyond their introduction papers. Therefore, we limited our choices to models that have participated in the ImageNet benchmark, evaluating the test datasets of ImageNet-1k or 22k, where data class names are not publicly available to prevent data leakage. We considered models with formidable accuracy for testing.

Lastly, the architectural designs were a critical consideration, as they represent a diversity of architectures in the evolving field of computer vision.

The choosen Model architectures are:

- RESNET
- CONVNEXT
- CVT
- VIT
- SWIN

Starting with the well-known ResNet, which is a pure convolutional neural network (convnet) model, we also consider ConvNeXt, a hybrid convnet model designed with transformer features; CvT (Convolutional Vision Transformer), a hybrid transformer designed with convnet features; ViT (Vision Transformer), a pure transformer good at capturing global context; and Swin Transformer, a new variant of Vision Transformer with state-of-the-art metrics. Swin Transformer models strong long-range dependencies, better grasps local relationships, and is significantly more efficient than ViT due to its window-based approach.

These diversified architectures will help us determine if there is a robustness trade-off concerning their performance range, interpretability (with ViTs having less explainability than convnets), computational costs, and depth. This is crucial for understanding their future applications in image classification.

We will test their robustness on multiple levels, including unseen structured data , and different types of adversarial attacks. We do not expect to find a clear winner, as there is no "one size fits all" model. Each of these architectures has its pros and cons.

ConvNeXt is known for its deep hierarchical representations, capturing intricate details in images. Its depth may be able to enhance the model's ability to generalize and detect subtle patterns in unseen data, which is essential for robustness testing.

ResNet's depth and capacity allow it to capture a wide range of features at various levels of abstraction. This capability is beneficial for handling complex datasets and can potentially improve robustness against attacks. Additionally, ResNet's skip connections facilitate smoother gradient flow during training, potentially mitigating the impact of adversarial perturbations.

ViT's attention mechanism enables it to model relationships between image patches effectively, capturing both local and global dependencies. This attention mechanism can contribute to the model's robustness by allowing it to focus on relevant features while ignoring irrelevant or adversarial ones.

Swin Transformers have demonstrated competitive performance on various tasks with significantly fewer computational resources compared to traditional transformers. These models can efficiently handle long-range dependencies, making them suitable for tasks requiring global context understanding. We expect to see strong results against image corruptions and perturbations.

CvT combines the advantages of CNNs and Transformers, potentially offering a better level of robustness to adversarial attacks than pure transformers. However, CvT is a less established architecture compared to others, and its performance might not match that of Swin Transformer or ViT.

2.3 Model Architectures

RESNET

ResNet, short for Residual Network, is a deep learning model architecture that revolutionized the field of computer vision. It was introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 paper titled "Deep Residual Learning for Image Recognition." [1]

The key feature of ResNet is it's residual connections, which address the problem of vanishing gradients that occurs in very deep models. In these circumstances the model's gradients can become extremely small during backpropagation, significantly decreasing the model's learning ability.

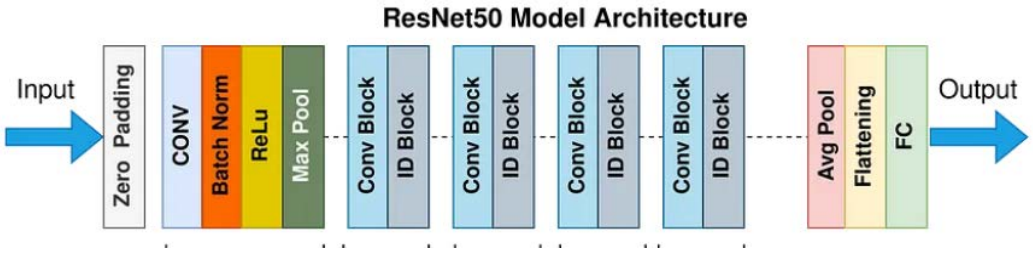


Figure 1.Resnet50 Model Architecture [2]

The basic architecture of a resnet model consists of :

Initial Convolutional Layer: The input image is passed through an initial convolutional layer with a large kernel size to extract basic features.

Stacked Residual Blocks: It's the core of the ResNet architecture and it consists of multiple blocks that can contain several residual units.

Downsampling Layers: Periodically, the spatial dimensions of the feature maps are reduced using techniques such as convolutional layers with a stride of 2 or max-pooling layers. This downsampling helps increase the receptive field and decrease computational cost.

Global Average Pooling: At the end of the convolutional layers, global average pooling is often applied to reduce the spatial dimensions of the feature maps to a vector of features. This operation helps to condense the spatial information into a compact representation.

Fully Connected Layer (Optional): In some architectures, a fully connected layer is added on top of the global average pooling layer to perform classification or regression tasks.

Output Layer: The final layer of the network produces the desired output, which could be class probabilities in the case of image classification or bounding box coordinates in the case of object detection.

The essence of ResNet lies in its residual blocks. These blocks allow for deeper networks without the vanishing/exploding gradient problem that plagued earlier their deeper models.

A residual block consists of two main components:

1. **Identity Mapping:** The input to the block (often referred to as the "identity") is passed through a series of convolutional layers, batch normalization layers, and activation functions. This sequence of operations is intended to learn a transformation that maps the input to some higher-level representation.

2. **Shortcut Connection:** In addition to the main path through the convolutional layers, a shortcut connection (also known as a skip connection or a residual connection) is introduced. This connection directly forwards the input to the output of the block, bypassing the convolutional layers. Mathematically, the output of the residual block is the sum of the output of the convolutional layers and the input to the block.

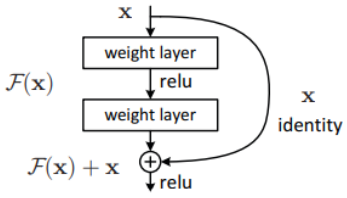


Figure 2.Resnet Residual Block [2]

ResNet architectures come in various depths, typically denoted as ResNet-{X}, where X represents the number of layers. For example, ResNet-50 consists of 50 layers, including convolutional layers, pooling layers, and residual blocks. There are also newer variants like ResNetV1.5 and ResNetV2. In our experiment, we are going to use the ResNet-50v2 variant, so it is appropriate to analyze this specific variant.

ResNetV2 is a newer variant of ResNet. The main difference in their residual blocks is that ResNetV2 applies Batch Normalization and ReLU activation to the input before the multiplication with the weight matrix (convolution operation). In contrast, ResNetV1 performs the convolution followed by Batch Normalization and ReLU activation.

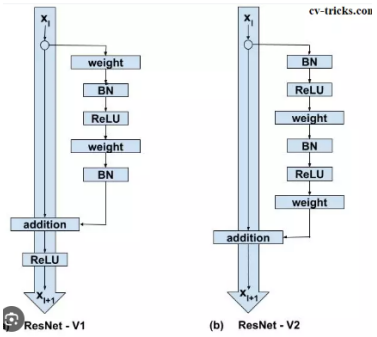


Figure 3.ResnetV1 Vs ResnetV2 [3]

Deployed Model Info:

Name: Resnet50_v2
Parameters: 23.56M
Layers:50
Weights: imagenet-1k
Imagenet Top-1 Accuracy: 75.3
Library: Keras
https://keras.io/api/keras_cv/models/backbones/resnet_v2/

ResNet has become a cornerstone in the field of deep learning and has been widely adopted in both research and practical applications due to its effectiveness and ease of implementation.

CONVNEXT

ConvNeXt, short for Convolutional Neural Network for vision Transformer, is a state-of-the-art image classification architecture known for its high accuracy and efficiency.

ConvNeXt falls into a category known as hybrid convolutional architecture, combining elements from both traditional Convolutional Neural Networks (CNNs) and Transformers. It was first proposed in the paper "A ConvNet for the 2020s" [4] during a time when Vision Transformers were at their peak, and ConvNets were losing ground in market share. Essentially, ConvNeXt is a modernized version of ConvNet, heavily inspired by Transformers.

The central architecture of the model is based on a resnet with features of the new techniques used mainly in vision transformers. Its successful new features can be summarized below:

New training techniques: incorporation of AdamW optimizer

Changing stem to "Patchify": The stem cell design is concerned with how the input images are processed at the network's beginning. It splits an image into a sequence of patches.

Improving the models Macro Design: Changing the stage ratio by adjusting the number of blocks in each stage.

Incorporating Depthwise Convolutions: A special case of grouped convolutions that reduce the number of floating-point operations (FLOPs) while widening the network. By mixing information in the spatial dimension, similar to self-attention in transformers, performance is significantly enhanced.

Implementing Inverted Bottlenecks: ConvNeXts make use of inverted bottlenecks this is done by creating an inverted bottleneck block that widens the Hidden dimension of the MLP block by four times compared to the input dimension,

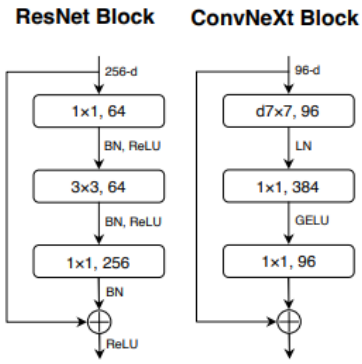


Figure 4.ResnetBlock Vs ConvNext Block [23]

Introducing Larger Kernel Sizes: To match the power of Vision Transformers with their global receptive field and self-attention, ConvNeXts adopt larger kernel sizes.

Making Micro Design Choices: Replacing the Rectified Linear Unit (ReLU) activation function with Gaussian Error Linear Units (GELUs) and reducing the usage of normalization layers aligns ConvNeXts more closely with transformer architecture.

ConvNeXt variants are designated as ConvNeXt-{size}, representing different implementations of the model's depth:

\

ConvNeXt-Tiny: Smallest and most efficient variant, suitable for resource-constrained environments.
ConvNeXt-Small: Offers a good balance between accuracy and efficiency.
ConvNeXt-Base: Larger variant with higher accuracy potential.
ConvNeXt-Large: Even larger model for pushing the limits of accuracy but requiring more resources.
ConvNeXt-XL: Largest variant achieving state-of-the-art performance.

Deployed Model Info:

Name: Convnext-B
Parameters: 89M
Layers:12
Weights: imagenet-1k
Imagenet Top-1 Accuracy: 83.8%
Library: keras

CVT (Convolutional Vision Transformer)

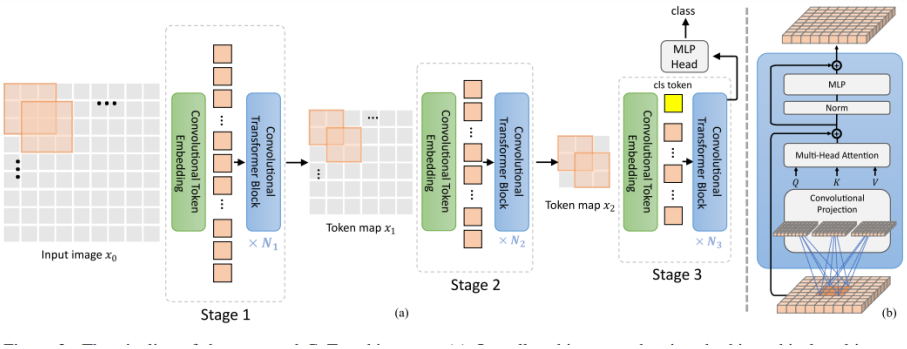


Figure 5.Cvt model Architecture [5]

The CvT (Convolutional Vision Transformer) is a hybrid transformer model architecture derived from CNN networks. It was proposed in the paper CvT: Introducing Convolutions to Vision Transformers [5] .

The idea behind CvT is to strategically introduce convolutions to the ViT (Vision Transformer) structure to enhance performance and robustness while maintaining high computational and memory efficiency.

CvT is a state-of-the-art model that, due to the built-in local context structure introduced by convolutions, no longer requires a position embedding. This feature gives it a potential advantage for adaptation to a wide range of vision tasks requiring variable input resolution.

Its architecture differs from a ViT through two core adaptations from CNNs: convolution projection and convolutional token embedding.

Initially, the Transformers are divided into several stages, creating a hierarchical arrangement. At the start of each stage, there's a convolutional token embedding step. Here, a convolution operation with overlap is applied to a token map reshaped into a 2D grid. Following this, layer normalization is applied. This setup enables the model to capture local details, gradually reduce the sequence length, and enhance the token features' dimensionality across stages. This process achieves spatial downsampling while increasing the number of feature maps, akin to how convolutional neural networks operate.

Secondly, in the Transformer module, the traditional linear projection before each self-attention block is replaced with a convolutional projection. This new approach involves applying a depth-wise separable convolution operation with dimensions s × s on a token map reshaped into a 2D grid. This modification enhances the model's ability to grasp local spatial context and diminish semantic uncertainty within the attention mechanism. Additionally, it enables better control of computational demands, as the convolution's stride can be used to downsample the key and value matrices, boosting efficiency by 4× or more with minimal impact on performance.

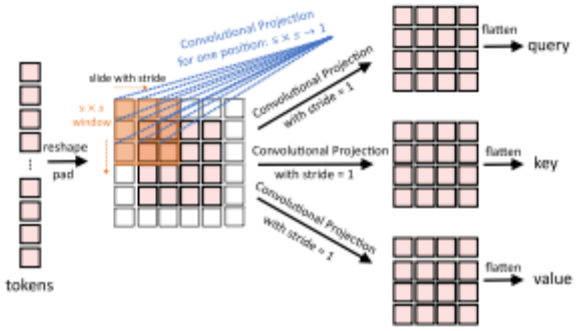


Figure 6.Convolutional projection

There are several variants of the CvT model, defined by differences in depth and width:

CvT-13 and **CvT-21:** These are considered basic models, with 19.98M and 31.54M parameters, respectively. The number represents the total number of transformer blocks in the model.

CvT-W24: A broader model with a larger token dimension per stage, denoted as CvT-W24 (W indicating Wide), resulting in 298.3M parameters.

Deployed Model Info:

Name: Cvt-21
Parameters: 32M
Layers:21
Weights: imagenet-1k
Imagenet Top-1 Accuracy: 83.0%
Library: transformers
Model trained by: microsoft
https://huggingface.co/microsoft/cvt-21

SWIN TRANSFORMER

The Swin Transformer is a type of Vision Transformer Proposed by Swin Transformer: Hierarchical Vision Transformer using Shifted Windows [6]. Its creation addresses the challenges of adapting Transformers from language to vision, considering the large variations in the scale of visual entities and the high resolution of pixels in images compared to words in text. The Swin Transformer features a hierarchical architecture with representations computed using shifted windows, achieving state-of-the-art results, especially with high-resolution images.

The Swin Transformer differs from pure ViT counterparts through its hierarchical representation architecture, where the number of tokens is reduced by patch merging layers as the network deepens. Additionally, it incorporates a Swin Transformer block that replaces the standard multi-head self-attention (MSA) module in a Transformer block with a module based on shifted windows.

Core Components of Swin Transformer:

1. **Patch Splitting:** The Swin Transformer begins by splitting an input RGB image into non-overlapping patches using a patch splitting module, similar to ViT. Each patch is treated as a "token," with its features set as a concatenation of the raw pixel RGB values.

2. **Hierarchical Representation:** Several Transformer blocks with modified self-attention computation (Swin Transformer blocks) are applied to these patch tokens.

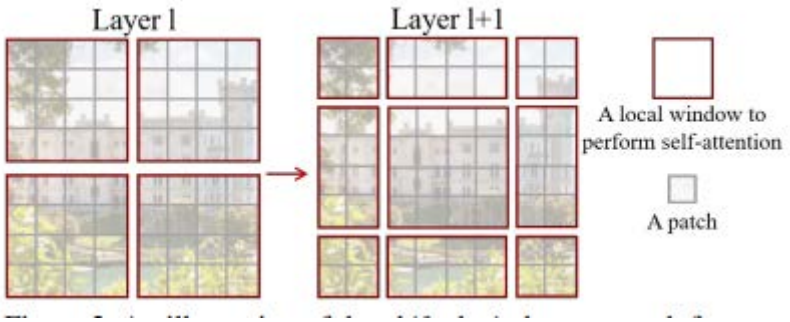
   The initial stage **(Stage 1)** maintains the number of tokens H/4 x W/4. To produce a hierarchical representation, patch merging layers reduce the number of tokens as the network gets deeper:

   Stage 2: Applies Swin Transformer blocks for feature transformation, reducing the resolution to H/8 x W/8

   Stage 3: Further reduces the resolution to H/16 x W/16.

   Stage 4: Final stage reduces the resolution to H/32 x W/32.

   -



Swin Transformer Block: The Swin Transformer block replaces the standard MSA module with a shifted window-based MSA module. Each block consists of:

- A shifted window-based MSA module.
- A 2-layer MLP with GELU nonlinearity.
- LayerNorm (LN) applied before each MSA module and MLP.
- A residual connection after each module.

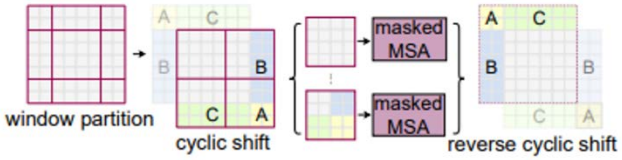


Figure 7.Swin Tranformer Block

Shifted Window Partitioning:

- The first module uses a regular window partitioning strategy starting from the top-left pixel, partitioning the feature map into non-overlapping windows.
- The next module shifts the window configuration by half the window size, introducing connections between neighboring non-overlapping windows from the previous layer. This approach enhances the model's ability to capture local and global context, effective in image classification, object detection, and semantic segmentation.

Architecture Variants:

- **Swin-B:** A model with size and computational complexity similar to ViT-B/DeiT-B.
- **Swin-T:** About 0.25× the size and complexity of Swin-B, similar to ResNet-50 (DeiT-S).
- **Swin-S:** About 0.5× the size and complexity of Swin-B, similar to ResNet-101.
- **Swin-L:** About 2× the size and complexity of Swin-B, pushing the limits of performance.

Deployed Model Info:

Name: swin- transformer

Parameters: 88M

Layers: 12

Weights: imagenet-1k

Imagenet Top-1 Accuracy : 85.2%

Library: transformers

Model trained by: microsoft

VISION TRANSFORMER

Transformer models debuted in the field of Natural Language Processing (NLP), revolutionizing the entire domain. Since then, many have attempted to apply their benefits to Computer Vision by incorporating some of their architectural principles into existing convolutional neural networks. The pure Vision Transformer (ViT) model came much later in this effort, as proposed in the paper “An Image is Worth 16x16 words: Transformers For Image Recognition At scale” [7] once again stunning the scientific community with new ways to interpret images.

The Vision Transformer (ViT) is a transformer encoder model (similar to BERT) trained in a supervised manner. Images are presented to the model as a sequence of fixed-size patches (resolution 32x32), which are linearly embedded. Just like transformers in NLP, a special token [CLS] is added to the beginning of a sequence to use it for classification tasks. Furthermore, absolute position embeddings are added before feeding the sequence to the layers of the transformer encoder.

The core architectural components of ViTs can be summarized as follows:

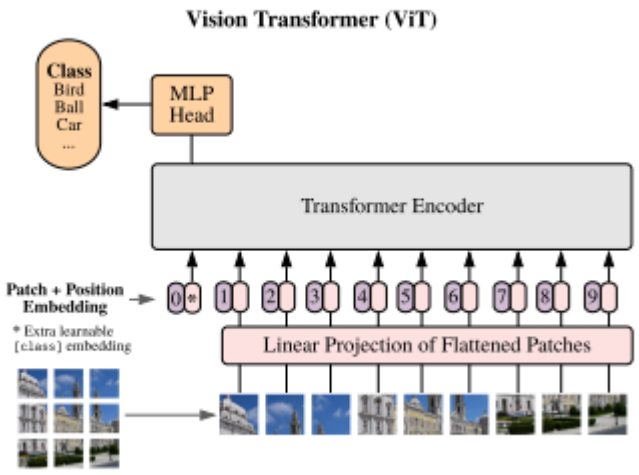


Figure 8.Vision Transformer Architectures [7]

Patchification of input image

Instead of analyzing the whole image at once, ViTs break it down into smaller, more manageable chunks called patches and then flattens them.

Patch Embedding:

Each patch is converted into a lower-dimensional representation called an "embedding" using a learned function ,positional embedding are added as well.This representation captures the local information within the sequences of patches.

Transformer Encoder
The encoded patches (fingerprints) are fed into a series of transformer encoder layers. Each layer allows patches to "communicate" and learn relationships with each other, regardless of their position in the image.

Transformer Encoder Stack:

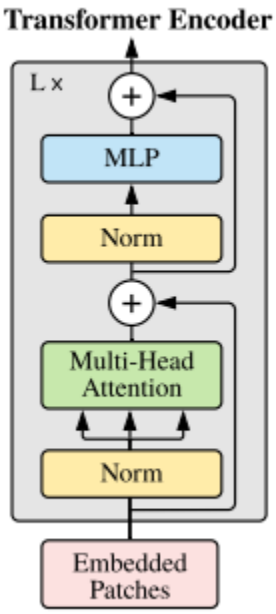


Figure 9.Transformer Encoder Block

The encoder block is identical to the original transformer The only difference is the number of these blocks. Each multi-head attention block comprises:

- **Three Linear Layers:** These layers process the queries, keys, and values independently.

- **Scaled Dot-Product Attention:** This function combines the queries, keys, and values to compute attention scores. The multi-head attention mechanism repeats these operations multiple times (h times, where h is the number of heads) and performs them in parallel. By using multiple heads, the network can learn different ways to attend to information, leading to richer representations.

- **The MLP** (multi-layer perceptron) size refers to a module of linear transformation layers. The embedding size is kept fixed throughout the layers to enable short residual skip connections.

ViTs have achieved state-of-the-art performance on various tasks, including image classification, object detection, and semantic segmentation. They often outperform CNNs, especially on large datasets and complex tasks. Unlike CNNs, which are limited by local receptive fields, ViTs can capture long-range dependencies between image elements thanks to their self-attention mechanism. This allows them to better understand the overall context and relationships within an image, leading to improved accuracy. The shared architecture between ViTs and their language processing counterparts (Transformers) opens doors for exciting possibilities, such as joint image-text understanding and generation.

However, ViTs also have limitations. While newer models are more efficient, training ViTs can still be computationally expensive compared to CNNs. Additionally, the explainability behind ViT predictions can be challenging due to the complex nature of self-attention mechanisms. Overall, ViTs represent a significant step forward in computer vision, offering improved performance, flexibility, and potential for future advancements. As the field evolves, we can expect further improvements in efficiency and explainability, making ViTs even more powerful and accessible.

Model Variants: ViT configurations are based on those used for BERT BERT (Devlin et al., 2019).
The “Base” and “Large” models are directly adopted from BERT, with the addition of a larger “Huge” model. Brief notation indicates the model size and the input patch size: for instance, ViT-L/16 means the “Large” variant with 16×16 input patch size. Note that the Transformer’s sequence length is inversely proportional to the square of the patch size, making models with smaller patch sizes computationally more expensive.

Deployed Model Info:
Name: google/vit-base-patch32-384
Parameters: 86M
Layers=12
Weights: imagenet-1k
Imagenet Top-1 Accuracy: 77.91%
Library: transformers
Model trained by: google
https://huggingface.co/google/vit-large-patch32-38

2.4 Datasets matter

Apart from the choice of the model itself, the process—especially if the problem is complicated—requires an abundance of data to train, hyperparameter tuning, and thorough evaluation. In this cycle, datasets play a crucial role in the model's ability to learn and generalize to its full potential.

In the realm of deep learning, the dataset is the cornerstone upon which the entire structure of neural network training and evaluation rests. In the domain of image classification, where intricate patterns and subtle features differentiate one object from another, the dataset assumes a paramount role. Its quality, diversity, and size profoundly influence not only the accuracy of the model but also its robustness in real-world scenarios.

A rich and diverse dataset facilitates the learning process by exposing the neural network to a myriad of variations and complexities present in real-world data. By encompassing a wide spectrum of objects, backgrounds, lighting conditions, and orientations, it equips the model with the versatility necessary to generalize effectively. Conversely, a limited or biased dataset can result in a narrow understanding, rendering the model susceptible to erroneous classifications when confronted with unforeseen circumstances.

Robustness, a coveted attribute in any machine learning model, hinges largely on the quality and comprehensiveness of the dataset. A robust classifier exhibits resilience in the face of adversarial examples, noisy data, or variations in input characteristics. This resilience is nurtured through exposure to diverse examples during the training phase, where the model learns to discern salient features amidst a variety of input signals.

However, the pursuit of robustness is not without its challenges and limitations. Despite meticulous curation, datasets inevitably encapsulate certain biases and idiosyncrasies reflective of their creators and sources. These biases, if left unaddressed, can permeate the neural network, perpetuating societal prejudices or misconceptions. Therefore, rigorous evaluation protocols, encompassing fairness metrics and bias detection mechanisms, are imperative to scrutinize the model's performance across different demographic groups and contexts.

Moreover, the evaluation of a neural network's efficacy transcends mere accuracy metrics. While accuracy serves as a basic measure of performance, it often fails to capture the nuances of model behavior, especially in scenarios where misclassifications have disparate consequences. Metrics such as precision, recall, F1 score, and area under the receiver operating characteristic curve (ROC-AUC) offer a more nuanced portrayal of the model's performance across different classes and imbalance scenarios.

2.4.1 Datasets involved

Discovering the robustness of our selected models against complex attacks is crucial to our research. In assessing the robustness of image classification models, it is imperative to scrutinize their performance across diverse datasets to ensure their generalizability and reliability in real-world scenarios. While adversarial attacks serve as a critical benchmark for evaluating model vulnerability, beginning with a comparative analysis across distinct datasets offers valuable insights into the inherent adaptability and transferability of these models.

In this thesis, we embark on a comprehensive examination of the robustness of image classification models, specifically those trained on the widely utilized ImageNet dataset, when confronted with the ObjectNet dataset. The rationale behind starting our investigation with a comparison across datasets lies in the acknowledgment of the intricate interplay between model architecture, dataset characteristics, and environmental factors in shaping model behavior.

ImageNet, renowned for its vast diversity and scale, has been a cornerstone in the development and benchmarking of image classification algorithms. Conversely, ObjectNet introduces a distinct set of challenges by presenting images captured in real-world settings, encompassing variations in lighting conditions, object occlusion, and viewpoint perspectives that may not be fully represented in the training data.

By subjecting these models to the ObjectNet dataset prior to adversarial attacks, we aim to discern their baseline performance and identify any inherent biases or limitations stemming from dataset-specific features. This preliminary evaluation serves as a critical precursor to adversarial testing, enabling a more nuanced understanding of model behavior and informing targeted strategies for enhancing robustness.

Moreover, by initiating our analysis with dataset divergence, we underscore the importance of robustness beyond the confines of specific training data distributions. Our approach aligns with the overarching objective of fostering resilient and adaptable image classification models capable of exhibiting consistent performance across heterogeneous environments and unforeseen challenges.

Through this research endeavor, we aim to contribute not only to the ongoing discourse surrounding model robustness but also to provide practical insights for the development and deployment of image classification systems in real-world applications where reliability and generalizability are paramount.

In the section below, we introduce the datasets involved in these experiments: the ImageNet dataset, which is the dataset the models were trained on, and the ObjectNet dataset, which they will be tested upon.

2.4.2 ImageNet

The ImageNet project is a large visual database designed for use in visual object recognition software research. More than 14 million images have been hand-annotated by the project to indicate what objects are pictured. In at least one million of these images, bounding boxes are also provided. ImageNet contains more than 20,000 categories, with a typical category consisting of several hundred images.

Figure 10.Imagenet Dataset sample [8]

The ImageNet dataset was conceived in 2009 to address the scarcity of large-scale labeled datasets. Since its inception, it has evolved into a monumental repository, marking the dawn of a new era in visual recognition. The fusion of deep learning methodologies and vast datasets has propelled the boundaries of visual recognition.
Over the years, ImageNet has undergone a remarkable journey of growth and refinement. From its early stages of manual annotation efforts to the advent of crowdsourcing techniques, the dataset has expanded in scale and diversity, encapsulating a rich tapestry of visual content.

The impact of ImageNet reverberates across the landscape of computer vision research, permeating diverse domains and applications. From object recognition and scene understanding to visual reasoning and beyond, the dataset serves as a litmus test for algorithmic prowess, fostering a culture of rigorous evaluation and benchmarking. On 30 September 2012, a convolutional neural network (CNN) called AlexNet achieved a top-5 error rate of 15.3% in the ImageNet 2012 Challenge, which was more than 10.8 percentage points higher than that of the runner-up. This achievement was a cornerstone in the deep learning revolution.

Nowadays, the ImageNet dataset is used both as a standard training dataset and as a benchmark test for the efficiency of many computer vision architectures. This is amplified by numerous benchmarking competitions, including the ILSVRC, which utilizes a subset of ImageNet for the task of image classification.

Subsets

There are various subsets of the ImageNet dataset used in different contexts. One of the most highly used subsets is the "ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012-2017 image classification and localization dataset." This subset is also referred to in the research literature as ImageNet-1K or ILSVRC2017, reflecting the original ILSVRC challenge that involved 1,000 classes.

ImageNet-1K contains 1,281,167 training images, 50,000 validation images, and 100,000 test images. Our models are trained with this subset and validated for their effectiveness using the validation images.

The full original dataset is referred to as ImageNet-21K. ImageNet-21K contains 14,197,122 images divided into 21,841 classes. Some papers round this number up and refer to it as ImageNet-22K.

Images Details

The images in the dataset have various resolutions, with the most common being 224x224, 256x256, 512x512, and 1024x1024 pixels. The aspect ratios of the images may also vary, reflecting different compositions and viewpoints captured by the original sources. The formats are mainly JPEG, but other formats exist as well, and their color spaces are primarily RGB, with some grayscale exceptions.

Challenges

Despite its monumental stature, ImageNet grapples with inherent biases, annotation inconsistencies, and the ever-evolving nature of visual semantics. These obstacles have spurred innovation, catalyzing efforts to enhance dataset quality and mitigate biases. Through

collaborative endeavors and community-driven initiatives, strides have been made to fortify the dataset's integrity, ensuring its continued utility as a benchmark for computer vision research.

2.4.3 ObjectNet

ObjectNet is a large real-world test set for computer vision with bias control, where object backgrounds, rotations, and imaging viewpoints are random. Fueled by the shortcomings of existing datasets like ImageNet, ObjectNet emerges as a beacon of realism, offering a diverse array of objects in natural settings.

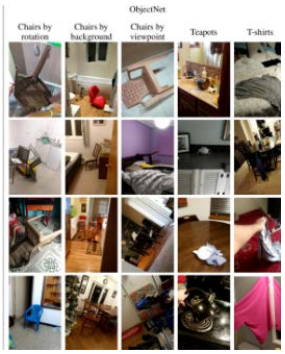


Figure 11.Objectnet images sample [22]

Unlike traditional datasets, ObjectNet takes a unique approach by intentionally lacking a training set. Instead, it provides a 50,000-image test set, similar in size to ImageNet, but without a corresponding training counterpart. This encourages generalization and tests models' ability to perform well without relying on prior training data. The dataset incorporates controls inspired

by scientific experiments, removing confounding factors to ensure that models cannot exploit trivial correlations in the data.

ObjectNet aims to represent real-world scenarios more accurately by including variations in object backgrounds, rotations, and imaging viewpoints. To achieve this, images are captured using crowdsourcing and are manually annotated . One of the defining features of ObjectNet is its emphasis on capturing everyday objects in natural settings, such as homes, offices, outdoor environments, and public spaces. Within each object class, ObjectNet strives to achieve a balanced representation to mitigate biases and ensure comprehensive coverage. This entails an equitable distribution of images and instances across various classes, avoiding skewness towards certain categories.

The ObjectNet test set contains 50,000 images categorized into 313 object classes. There is a partial overlap between ObjectNet and ImageNet, with ObjectNet containing 113 object classes also present in ImageNet. The format of the images in ObjectNet is not explicitly mentioned in any available sources, but it is likely standard JPEG (.jpg) or Portable Network Graphics (.png) format. The images come in various resolutions and their color space is usually RGBA, RGB, or grayscale.

Due to ObjectNet's purpose and its variations, the representation in its classes is a perfect candidate to start our robustness journey. Many image classifiers, including some of our model variants , have been tested on ObjectNet and experienced a significant 40-45% drop in performance compared to their performance on other benchmarks.

3 Computer Vision Robustness Experimental Setting

In this experiment, we investigate the robustness of neural networks in image classification tasks, particularly focusing on the transferability of models trained on the ImageNet dataset to the ObjectNet dataset. Our primary objective is to assess how well neural networks generalize when trained on ImageNet and subsequently tested on ObjectNet, scrutinizing their performance under different conditions to uncover potential vulnerabilities and strengths.

To initiate our investigation, we designed an experimental framework for assessment in both the ImageNet and ObjectNet datasets.

Project Details:

- Python 3.11
- Jupyter Notebook
- Libraries : Tensorflow/Keras and Transformer , sklearn, numpy, torchvision
- Files: ObjectNetExperiment.ipynb , ImagenetModels.ipynb, functionsFile, DatasetLabels
- GPU: Nvidia Geforce 3050
- Github: https://github.com/tzimk323/computerVision

Each of our model assessments took place within the respective ObjectNet and ImageNet files. Our methods framework is detailed in the `functionsFile`, and the `DatasetsLabels` file was used to create the ground truth data for both datasets.

Our framework is split into two core function types: methods to assess the models of the TensorFlow-Keras library (ResNet and ConvNeXt) and methods to assess the transformer models (Swin-Transformer, ViT, and CvT21). The assessment methods were designed to be applicable to all the models within each library type and for both the ObjectNet and ImageNet datasets. Consistent preprocessing was applied to both datasets to ensure the reliability of our results.

3.1 Imagenet Dataset:

Before subjecting our models to the ObjectNet dataset robustness assessment, we needed to establish a performance baseline due to the different model architectures, configurations, and training types. This baseline would provide concrete data on performance decreases. To achieve this, we first assessed our models using their training dataset's validation sample.

The validation sample was taken from the ImageNet-1K validation images, which, like ObjectNet, contains at least 50,000 images. For time and GPU resource efficiency, we tested our models on a smaller sample of 15,000 images. This sample was collected from the initial validation set with stratification to ensure the target classes were proportionally represented.

From the ImageNet project, we aligned our images' true prediction indices by the datasets' image names. The images were preprocessed according to each library's model built-in preprocessor, which included rescaling, resizing to 224x224, and normalization of the images. We also transformed all the images' color spaces to RGB. The evaluation metrics used were accuracy and F1 score, which accurately represent the results for image classification.

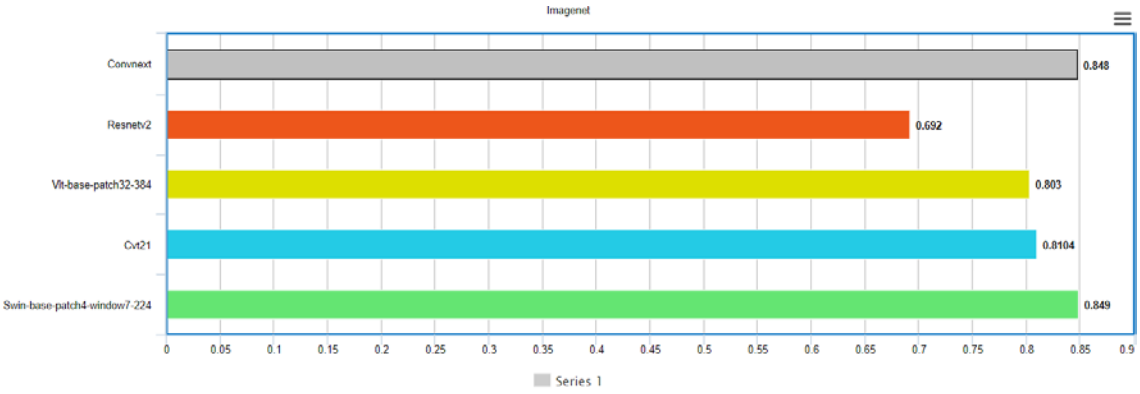


Figure 12.Imagenet Baseline Experiment Results

Except for ResNet, which is an industry standard model but not state-of-the-art, all other models scored above 80% accuracy, with the Swin Transformer achieving the best results at 0.849, slightly ahead of the ConvNeXt model at 0.848. The CvT21 and Vision Transformer models followed with scores of 0.8104 and 0.803, respectively.

3.2 Objectnet Dataset:

To assess our robustness results on the ObjectNet dataset, the first challenge we faced was to create the true prediction values. The dataset images are divided into folders, each containing a certain class. The overlapping classes of ObjectNet and ImageNet are 113 out of the total 313, so we had to extract only their intersection. Additionally, some overlapping classes were not unique; for instance, ImageNet might have a general class such as "alarm clock," while ObjectNet could distinguish between "analog clock" and "digital clock." To get the true annotations for every image, we mapped each folder name to its annotation class name and its ImageNet counterpart. To handle multiple class mappings, we developed an evaluation algorithm that considers a prediction correct if one of the ObjectNet classes is a subset of the ImageNet class.

After mapping the annotations and ImageNet classes, we used the same prediction methods as for ImageNet, taking extra considerations when different image formats or color spaces (RGBA) were present, which were absent in the ImageNet dataset. The evaluation process for the entire dataset for our five models took at least 3 hours per model, and the results were saved in a text file along with their respective indices.

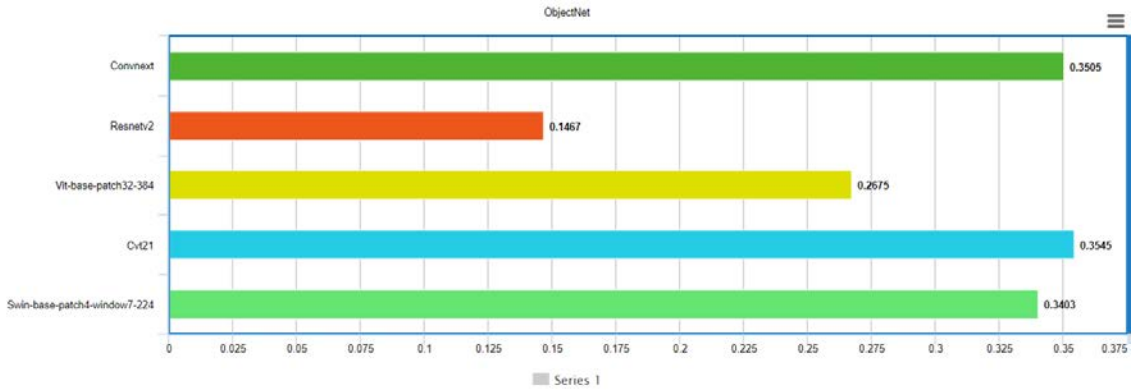


Figure 13.ObjectNet experiment results

As we can see from bar chart the all the results holds huge differences in their accuracy metrics.

3.3 Results Analysis

Our experiments reveal a concerning phenomenon: all five networks exhibited significant performance drops compared to their reported accuracy on our baseline benchmark. This raises critical questions about the generalizability and real-world applicability of these models.

- Cvt drop=56.24%
- Convenext drop=58%
- Swin drop=59.95%
- Vit drop=66.7%
- Resnet Drop=78.79%

There are several potential explanations for the observed performance drops.

Architectural limitations

The performance drops affect all our architectures, but not uniformly. The worst performance came from the ResNet model, which is one of the oldest and simplest architectures used in our experiment, followed by the ViT classifier. The best performances were derived from the two hybrid models, followed by the Swin Transformer. Given our results, it is safe to assume that architectural limitations are evident. The residual blocks of ResNet and the multihead attention mechanism of ViT are great feature extractors, but they seem insufficient by themselves to generalize well for the complexity and diversity of the ObjectNet dataset compared with the data they were trained on. Although the hybrid models' results seem more promising by combining traits of more than one architecture, they too lack the ability to generalize in a manner suitable for real-world applications.

Dataset Bias

Training a model on a complex dataset with more diversity is crucial for our network to generalize its knowledge. Showing only specific types of objects from specific angles causes the model to develop bias about what the object is. The nature of an image in 2D, as opposed to the 3D world, should be considered to reveal the true form of an object for our models to understand, along with the different shapes and diversities an object could have.

Overfitting

Displaying an object in a certain frame holds the possibility of the model finding irrelevant features, causing the model to overfit on the frame itself and not the object in question.

Preprocessing

We used specific preprocessing methods for the prediction paradigms, which are the same methods the models were trained on. However, the representation of a preprocessed image can also be a dataset-specific statistical tool to help the model capture more object features. For example, if the trained dataset's images use more bright colors, the preprocessing could lower the brightness of these images to make their features more robust. In a dataset where the brightness is very low, the results of such a filter could have the opposite outcome. This phenomenon is very clear in the ObjectNet dataset compared to ImageNet in relation to the objects' positions and the rescaling of the models' preprocessing methods according to the images' center.

Annotation Differences

The ObjectNet dataset has 113 classes that overlap with our trained models' dataset, but this definition is incomplete. Some ObjectNet classes can only be subsets or approximations of ImageNet’s corresponding ones, which could lead the model to fail to make a correct prediction. Some might argue that this phenomenon is specific to this dataset, but this is not the case. Our definitions in human languages (and annotations) to name entities we see can sometimes be abstract, leaving us to understand a concept only by context. However, it is not only our language limitations that we could blame for the vast diversity of the real world. A deep learning model trained on a dataset could learn some classes' features, but as it tries to optimize this process, it could learn to identify them in relation to others, which could be considered the model's own context of the world.

4 Vulnerabilities in Computer Vision

In the previous chapter, we explored the robustness of AI models in computer vision by evaluating their performance on the challenging ObjectNet dataset. While such evaluations highlight the model's ability to handle variations in real-world data, ensuring robustness goes beyond simply addressing mismatched distributions. The concept of robustness in AI extends beyond mere accuracy metrics, encapsulating the system's ability to maintain functionality and reliability in the presence of unforeseen circumstances and malicious interventions. In safety-critical applications and security-conscious environments, AI systems can be vulnerable to deliberate manipulation. This chapter delves into the crucial aspect of robustnes and resilience against adversarial attacks. Through our investigation, our main goal is to shed light on this emerging attack paradigm and provide a comprehensive understanding of robustness in computer vision, encompassing not only data variations but also intentional attempts to exploit model weaknesses that could affect the model's performance and the security and privacy of the entire system.

Vulnerabilities in Software systems

Vulnerabilities in software systems refer to their weaknesses or flaws in the design, implementation, or configuration that can be exploited by attackers to compromise the integrity, confidentiality, or availability of the system. These vulnerabilities can exist at various levels, including the application layer, operating system, network protocols, and even in the hardware components. Their key aspects could include their types as they can manifest in different forms, their causes that they arose, and their impact on the targeted system.

4.1 Vulnerabilities in ML systems

Machine learning systems diverge from conventional algorithms and computer programs in their methodology. They acquire knowledge and formulate decisions by analyzing data obtained from their operational context. As a result, these systems exhibit distinct vulnerabilities, are susceptible to unique threats, and are susceptible to a different array of attacks compared to traditional computer systems .

In contrast to conventional software programs, the behavior of a machine learning model is derived from the data it undergoes training with. This fundamental difference results in four significant implications that, in comparison to traditional software systems, expand its attack surface and introduce novel types of vulnerabilities.

Firstly, since the behavior of the model is learned from the training data, the information contained within the training data becomes inherently ingrained in the machine learning model and consequently, in its predictions. This implies that the confidentiality of the training data and its data sources could be compromised through the machine learning model and its predictions, even if robust encryption and secure storage mechanisms are employed to safeguard the training data.

Secondly, the reliance on data for learning means that a machine-learning model can be compromised by attacking its training data or data sources. Any compromise, such as integrity loss, to these assets prior to training will be transferred to the model during the training process.

Thirdly, machine learning models pose challenges in terms of verification. Unlike traditional software libraries, deciphering the code of a machine learning model and identifying potential flaws and threats is complex. This complexity is exacerbated by the non-deterministic decision-making of some models. While input-output-based validation of machine learning models can offer statistical evidence of their anticipated functionality, these validation results cannot ascertain their correctness for all potential inputs. Consequently, this introduces risks associated with supply chain attacks, as third-party machine learning models are often utilized either in their original form or as a foundation for training other models.

Lastly, detecting adversarial data inputs during both training and inference stages presents a formidable challenge. Machine learning is employed precisely because establishing explicit rules to model certain data or phenomena straightforwardly is often impossible. As a result, there are typically no readily available and scalable methods to determine whether a given input data is benign or malicious. Additionally, inputs are sourced from the environment in which the machine learning system operates or from its users, and such input spaces cannot be narrowly defined. This

characteristic also means that attackers can compromise machine learning systems by manipulating their environment or influencing certain users.

The implications outlined above highlight why machine learning systems present distinct vulnerabilities compared to traditional software systems, necessitating a different approach to security assessment and implementation. An overview of the main vulnerabilities that machine learning systems can hold is summarized below:

***Model Poisoning**:* This attack alters a machine learning model by tampering with its training data or process. By injecting malicious data inputs during training, the attacker distorts the model's classification accuracy, rendering it impractical for real-world use.

***Model Evasion**:* Commonly targeting the inference phase, evasion attacks aim to manipulate the model's predictions by introducing carefully crafted malicious inputs, known as **adversarial examples.** These attacks seek misclassifications while minimally modifying the input, such as bypassing network intrusion detection systems.

***Model Stealing**:* These attacks compromise the confidentiality and intellectual property of a machine learning model during inference. Attackers exploit the information leaked by the model's query/response interactions to reconstruct a similar model, compromising the original model's confidentiality.

Training Data Inference: These attacks exploit information leaked by machine learning systems to compromise the confidentiality of training data and threaten individual or organizational privacy. Two main types exist: membership inference and model inversion attacks.

***Membership Inference*:** This attack seeks to determine if a record is part of the training data by observing the model's behavior with known and unknown inputs. It can reveal sensitive personal data like purchase records or medical history.

***Model Inversion*:** Here, attackers with partial knowledge of a data record try to infer missing attributes by querying the model with different possible values. The output analysis reveals the true attribute value present in the training data.

Supply Chain Vulnerabilities*:* External components, libraries, and code pieces used in machine learning systems can be compromised, leading to attacks on the resulting model or system. Verifying the integrity of these components is crucial to prevent supply chain attacks, especially those targeting specific ML components like training libraries and pre-trained models.

Deployment Vulnerabilities: Utilizing external cloud services for training and deploying ML models can introduce security risks if these platforms are compromised. Even with secured components and algorithms, a compromised platform can replace them with malicious ones, jeopardizing the entire system's security. Verifying the integrity of training and deployment platforms is essential to mitigate such risks.

Given the complexity and evolving nature of machine learning systems, it becomes imperative to fortify them against emerging threats and vulnerabilities. As outlined in the preceding discussion, the unique characteristics of machine learning models, such as their reliance on training data and their susceptibility to various forms of attacks, underscore the critical need for robust defenses.

By comprehensively understanding the attack surface and vulnerabilities inherent in machine learning systems, stakeholders can develop targeted defense strategies to mitigate potential risks effectively to address a more robust approach of these systems.

4.2 Vulnerabilities in computer vision

In the preceding chapter, we explored the vulnerabilities present in AI systems, highlighting the critical importance of addressing these vulnerabilities to ensure the reliability, safety, and security of AI applications. In this chapter, our focus narrows to the domain of computer vision, a subfield of machine learning systems that can inherit many of its vulnerabilities.

By enabling machines to interpret and understand visual information, computer vision systems have become integral to modern technology. Their range of applications can vary greatly across numerous fields, from autonomous vehicles to healthcare diagnostics. The study of their robustness is paramount, as in many cases, their decision-making could be a matter of life or death. This is why computer vision is a central focus when discussing vulnerabilities; it may be the most susceptible and the most examined field due to its corresponding exploitation.

However, simply identifying vulnerabilities is only half the battle. To effectively address them and build robust computer vision systems, we need a systematic approach. This is where taxonomies come into play. A well-defined taxonomy acts as a map, categorizing different types of computer vision tasks and their inherent vulnerabilities. By understanding how vulnerabilities manifest within each category, we can develop targeted mitigation strategies and assess their impact on the overall robustness of the domain.

4.3 Taxonomy of computer vision attacks

The vulnerabilities plaguing computer vision systems can be exploited through various attack vectors. To understand these effectively, we can categorize them into two primary domains: physical attacks and digital attacks. This distinction hinges on the method of manipulating the input data fed to the CV system during their real-world applicability.

Physical Attacks: These attacks manipulate the physical world surrounding the camera or sensor that captures the visual data. This manipulation aims to distort the data in a way that steers the CV system towards a desired, yet incorrect, output.

Digital Attacks: In contrast, digital attacks modify the digital representation of the visual data itself. This can involve introducing subtle, imperceptible modifications to images or videos before they are fed to the CV system.

While both main types of attacks have their own limitations, depending on the model and the type of access the attacker has, they can both be equally dangerous for our computer vision models. Physical attacks can disrupt the decision-making process through methods such as adversarial patches (like stickers), camouflage, and lighting manipulation. These can result in significant robustness issues. However, their effectiveness is highly correlated with the attacker's access to the specific model and its environment. In our research, we choose to analyze and test our models' robustness primarily based on digital attacks, taking into account their potential transferability, which could pose a broader threat.

Transferability refers to the ability of an attack crafted to fool one computer vision model (source model) to also deceive other models (target models) without any modifications. This poses a significant security risk, as attackers only need to generate one attack example against a single model and can potentially deploy it against a wide range of systems.

The transferability of digital attacks is not a universal trait, as many digital attacks have limited transferability, such as:

Injection Attacks: Inserting malicious code or manipulating existing code within the CV system itself to alter its behavior.

Poisoning Attacks: Contaminating the training data used to develop the CV system with manipulated images, leading the model to learn faulty patterns.

Compression Attacks: Exploiting compression artifacts introduced during image or video compression to introduce noise or manipulate the data subtly.

The highest possibility about an attacks transferability rate can be seen in the most studied computer vision attacks which are the adversarial attacks.

Adversarial attacks represent a sophisticated form of digital attack specifically designed to be imperceptible to the human eye. These attacks introduce subtle yet strategically crafted modifications to the input data, causing the CV system to produce a desired, yet incorrect, output. Adversarial attacks are particularly concerning due to their transferability, meaning they can often fool different computer vision models, and their targeted nature, where they aim to achieve specific misclassifications. The variants of these attacks can be classified by many factors such as:

Adversarial specifivity

Targeted: Targeted attacks aim to craft an adversarial image in order to lead the model to misclassify it in a predetermined class, chosen beforehand by the attacker.

Indiscriminate: On the other hand, in untargeted attacks, the attacker just seeks to fool the model by aiming any class different from the legitimate class corresponding to the original example.

Attackers Knowledge

White-box attacks:

In a white-box attack, the attacker has fully access to the model's and even the defense's parameters and architectures, whenever such defense exists.

Black-box attacks:

In this scenario, the attacker neither has access nor knowledge about any information concerning the classification model and its defense methods, when present.

Grey-box attacks:

In grey-box attacks, the attacker has access to the classification model, but does not have access to any information concerning its defense methods.

Perturbation Scope:

Universal-scoped perturbations: universal-scoped perturbations are image-agnostic perturbations, i.e. they are perturbations generated independently from any input sample

Individual-scoped perturbations They are generated individually for each input image;

Perturbation Visibility:

Optimal perturbations: these perturbations are imperceptible to human eyes

Indistinguishable perturbations: indistinguishable perturbations are also imperceptible to human eyes, however they are insufficient to fool deep learning models;

Visible perturbations: perturbations that, when inserted into a image, are able to fool deep learning models. However they can also be easily spotted by humans

Physical perturbations: are perturbations designed outside the digital scope and physically added to real-world objects themselves

Fooling images: perturbations which corrupt images to the point of making them unrecognizable by humans. Nevertheless, the classification models believe these corrupted images belong to one of the classes of the original classification problem.

Perturbation Type:

Additive Perturbations: The attacker adds a small amount of noise to the input image.

Multiplicative Perturbations: The attacker scales the input image by a small factor.

Adversarial Patches: The attacker adds a small patch to the input image that causes the model to misclassify it.

Attack Computation:

The algorithms used to compute perturbations can be **sequential and iterative**

The sequential algorithms compute in just one iteration, the perturbation that will be inserted into a legitimate image.

Iterative algorithms, in turn, make use of more iterations in order to craft the perturbation.

Attack Approach:

Adversarial attacks can also be organized with respect to the approach used by the attack algorithm to craft the perturbation.

Gradient-based attacks**:** these attacks approaches are the most used in literature. The gradient-based algorithms make use of detailed information of the target model concerning its gradient with respect to the given input. This attack approach is usually performed in white-box scenarios, when the attacker has full knowledge and access to the targeted model.

Transfer/Score-based attacks: the algortihms used either depend on getting access to the dataset used by the targeted model or the scores predicted by it in order to approximate a gradient.

Decision-based attacks: these are considered by the authors as a simpler and more flexible approach, since they require fewer changes in their parameters than gradient-based attacks. A decision-based attack usually queries the softmax layer of the targeted model and, iteratively, computes smaller perturbations by using a process called rejection sampling.

Approximation-based attacks: attacks that are based on this approach try to approximate a gradient for some targeted model or defense formed by a non-differentiable technique usually by applying numerical methods. These approximated gradients are then used to compute adversarial perturbations.

Although there are much more taxonomy types in the corresponding literature to name them all except the most frequenly used is out of this thesis scope. The taxonomy of adversarial attacks and the factors that help us divide them plays a crucial role in guiding the selection of algorithms for testing the robustness of machine learning models across various use cases.

4.4 Attack Strategies and their Selection

The algorithms used to generate adversarial perturbations are generally optimization methods that exploit generalization flaws in pretrained models to craft and insert perturbations into legitimate images. The realm of adversarial attacks in computer vision is an ongoing arms race. Attackers continually devise new methods to exploit the vulnerabilities of image recognition models. These vulnerabilities can stem from the inherent complexities of how models process visual information and make classifications. To succeed, attackers develop algorithms that

strategically manipulate input images in ways that cause the model to fail. In such context there are many algorithms that have been evaluated for promiment adversarial attack usage, with the most common classes being:

Gradients based attacks:

These attacks leverage the gradients of the model's loss function with respect to the input data. By analyzing how small changes to the input affect the model's predictions, these attacks craft adversarial examples by perturbing the input in the direction that maximizes the loss, thus leading to misclassification.

Optimization-Based Attacks:

Optimization-based attacks treat the generation of adversarial examples as an optimization problem. They iteratively adjust the input to minimize an objective function that represents the misclassification while constraining the perturbation magnitude. These attacks typically employ optimization algorithms to find the optimal perturbation.

Evolutionary Algorithms:

Evolutionary algorithms simulate the process of natural selection to find effective adversarial perturbations just as optimization-based attacks. They maintain a population of candidate adversarial examples and iteratively evolve them over multiple generations, favoring individuals that cause misclassification. Through selection, crossover, and mutation operations, these algorithms gradually improve the adversarial examples.

GAN-Based Attacks

These attacks utilize Generative Adversarial Networks (GANs) to generate adversarial examples. The attacker trains a GAN to produce images that the target model misclassifies. By iteratively training the GAN to generate more convincing adversarial examples, the attacker obtains images that are visually similar to natural images but lead to misclassification.

The corresponding experintal chapter delves into the inner workings of two prominent classes of attacks the Gradient -based using the Fast Gradient Sign Method (FGSM) and the Evolutionary-based attacks using a common genetic algorithm that will be implemented to evaluate or models reliability.

FGSM

The Fast Gradient Sign Method (FGSM) stands out as a fundamental technique in the realm of adversarial attacks, celebrated for its simplicity and efficacy. It falls into the category of white-box attacks, meaning the attacker has full access to the target model's architecture, parameters, and gradients. This knowledge empowers FGSM to create highly precise manipulations.

At its core, FGSM relies on gradients, which indicate the direction of greatest increase or decrease in a multi-dimensional space. In adversarial attacks, FGSM calculates the gradient of the model's loss function concerning the input image. This loss function measures how "incorrect" the model's prediction is for a given input. By computing the gradient, FGSM identifies the direction to modify the input image to maximize the loss function, pushing it towards a region where the model is likely to misclassify it.

The essence of FGSM lies in its manipulation strategy. Instead of directly applying the calculated gradient, which could result in significant and unrealistic changes to the image, FGSM utilizes a scaled version of the gradient's sign. By incorporating the sign, FGSM ensures the modification moves the image in the correct direction to maximize the loss, even if the magnitude of the change is reduced. This approach often yields subtle yet effective adversarial examples that may go unnoticed by humans but can significantly alter the model's prediction.

FGSM's simplicity and efficiency make it a valuable tool for initial robustness assessments. It provides a strong starting point to evaluate a model's susceptibility to adversarial manipulations and serves as a benchmark for assessing the effectiveness of more intricate attacks and a test for its transferability to the other models

Evolutionary-Based Attacks: Mimicking Real-World Adversaries

Evolutionary-based attacks represent a more advanced category of adversarial attacks, typically falling under black-box attacks. Unlike FGSM, which relies on complete knowledge of the target model, evolutionary attacks operate with limited information, mirroring real-world scenarios where attackers may only have access to the model's input and output behavior.

The core principle of evolutionary attacks draws inspiration from biological evolution. The attack begins with a seed image, which can be the original image or a slightly modified version. This seed image undergoes iterative refinement through a process resembling natural selection. A population of candidate adversarial examples is generated by applying small, random modifications to the seed image. Each candidate example is evaluated by feeding it into the target model, and successful manipulations are chosen for further modification in the next generation, while unsuccessful ones are discarded.

Over multiple generations, this iterative process allows the population of adversarial examples to evolve, gradually converging towards variants that consistently deceive the target model. The strength of evolutionary attacks lies in their ability to adapt and refine their strategies based on the model's response, making them effective against complex models with intricate decision boundaries and potentially transferable to models the attacker has not directly interacted with.

FGSM as a Benchmark: As a well-established and computationally inexpensive attack, FGSM offers a standardized method to evaluate a model's base-level vulnerability to adversarial examples. It serves as a point of reference to gauge the model's resilience against more sophisticated attacks. Also due to its simplicity could be easily replicated and transferred to other model architectures.

Evolutionary-Based Attacks Challenge Complexity: By mimicking a real-world attacker's approach, evolutionary attacks represent a more potent threat scenario. Their ability to adapt and exploit the model's weaknesses provides a more comprehensive assessment of the model's robustness in practical situations.

5 Computer Vision Vulnerabilities Experiments

5.1 FGSM experimentation setting

By incorporating both FGSM and evolutionary attacks into our robustness testing methodology, our research can paint a comprehensive picture of the target model's susceptibility to adversarial manipulations. This multifaceted evaluation will not only reveal their strengths and weaknesses in robustness comparative to the other architectures but also provide valuable insights into potential defense mechanisms that can be implemented to fortify the model against adversarial attacks in the real world.

Given Imagenet’s widespread adoption as a benchmark for assessing model robustness, conducting FGSM attacks on ImageNet-1k provides several compelling advantages.

Firstly, the sheer scale and diversity of ImageNet-1k ensure that the resulting adversarial examples generated through FGSM encompass a broad spectrum of visual attributes and semantic intricacies. This diversity is pivotal for evaluating the generalizability of model defenses against adversarial perturbations across a myriad of real-world scenarios and object categories.

Furthermore, ImageNet-1k's historical prominence and extensive adoption within the computer vision community lend credence to the relevance and applicability of FGSM attacks conducted on this dataset. Leveraging a dataset with established benchmark status facilitates meaningful comparisons with prior research efforts and enables the extrapolation of insights garnered from robustness testing to broader contexts within the field of computer vision.

Like our previous robustness baseline experiment , we did our testing on the validation set of imagenet-1k and for time and gpu resources efficiency we tested our models on a smaller sample of 7500 images which was collected from our initial sample with stratification to ensure the target classes proportion is not imbalanced.

Structure

Each of our model FGSM attack assessment took place in a ipynb file the *AdversarialFGSM .ipynb* along with its corresponding functions which were placed in *functionsFile.ipynb*

Due to our already made framework from our previous robustness baseline testing with imagenet-1k many of its features were used again when we saw fit, these features are:

- Metric functions
- Preproccesing
- Imagenet Groundtruth labels extraction
- Tensorflow-keras and Transfromers libraries each corresponding to our selected models

The most promiment change that was added to our experiment was a new function superset of our previous baseline prediction experiment method with two extra processes imported, the implementation of the FGSM attack on the picture to be predicted, and a method to save each of the adversary features for future transferability testing.

Both new processes were differently implemented according to each of our two main libraries Tensorflow-Keras and Transformers.

Tensorflow-Keras Implementation

FGSM attack in keras was implemented with the help of GradientTape along with GradientTape.watch and tf.sign

GradientTape is a TensorFlow utility that allows automatic differentiation of TensorFlow operations. It's particularly useful when you need to compute gradients with respect to some variables during the training of a neural network or any other differentiable computation.

tf.sign is a TensorFlow function that computes the sign of a tensor element-wise. It returns **-1.0** if the element is negative, **0.0** if it's zero, and **1.0** if it's positive. It's often used in custom loss functions or in conjunction with gradient descent algorithms.

tape.watch is a method of **tf.GradientTape** that explicitly watches a tensor so that gradients can be computed with respect to it, even if it's not a trainable variable.

By default, **tf.GradientTape** only tracks variables.

```
def create_adversarial_pattern(input_image, input_label,model):
  loss_object = tf.keras.losses.CategoricalCrossentropy()
  with tf.GradientTape() as tape:
    tape.watch(input_image)
    prediction = model(input_image)
    loss = loss_object(input_label, prediction)
  gradient = tape.gradient(loss, input_image)
  signed_grad = tf.sign(gradient)
  return signed_grad
```

After the creation of our adversarial patterns we add them to our processed image after being multiplied with the epsilon variable.

```
adv_x = img_array + eps * perturbations
```

To save the results for each image and to verify our perturbation implementation we return our preprocessed image to its original state before the preprocessing along with its perturbations and we save it.

Figure 14.Original Imagenet image

Figure 15.Same Image with Resnet152 FGSM pertubations

Transformers Implementation

To perform the FGSM attack on the models using the transformer library, we utilized the corresponding features of the PyTorch library. After obtaining the predicted probabilities of the image and its affected gradients, we used the torchvision library, specifically the .grad and torch.sign tools, to calculate the gradients and accumulate their signed values. We then added our perturbations to the image to be predicted, multiplied by our selected epsilon value.

```
input_image = preprocess(image).unsqueeze(0)
input_image.requires_grad = True
outputs = model(input_image)
probabilities = torch.nn.functional.softmax(outputs.logits, dim=-1)
top_probability = torch.max(probabilities, dim=-1).values
top_probability.backward()
gradients = input_image.grad
signed_gradients = torch.sign(gradients)
```

Due to insufficient documentation on each model's preprocessing process, we saved only the perturbations of the images in a text file. This approach allows us to add the same perturbations to the corresponding images when they are about to be predicted by another model.

Parameters Selection

The most critical parameter that was to be considered through our current experiment was the selection of epsilon(ε) value.

In the Fast Gradient Sign Method (FGSM) attack, epsilon (ε) is a parameter that controls the magnitude of the perturbation added to the input data. It represents the maximum allowable perturbation in each pixel or feature of the input image.

Selecting the appropriate epsilon value for an FGSM (Fast Gradient Sign Method) attack depends on various factors such as the specific model architecture, the nature of the dataset, and the desired level of perturbation.

Due to our various models architectures the complixity and the diversity of our dataset classes the final selection of our epsilon parameter was multifactored.

Models sensitivity

Many of our models have more complex architectures so for the perturbations to be able to affect their performance should be considerable higher than the more simple architectures.

Human detection threshold

After adding perturbations to our images, although visible to the human eye, we wanted to be able to be detectable by our own vision for bigger explainability in our experiment. Larger perturbations may affect some images beyond recognition even to us.

Trial and error

We tried different values in small samples to be able to conclude our final selection, with small perturbation values some more complex models in our current dataset did not seem to be affected nearly at all.

The value we chose was 0.15 which is considered high compared to previous research standards for such datasets.

5.2 Results

In this chapter, we delve into the crucial aspect of assessing the robustness of computer vision models against adversarial attacks, with a primary focus on the Fast Gradient Sign Method (FGSM) attack. The ability of machine learning models to generalize well to unseen data is paramount in real-world applications. However, the vulnerability of these models to adversarial examples, imperceptibly perturbed inputs crafted to deceive them, poses significant challenges to their reliability and security.

After implementing the FGSM attack on imagenet-1k to all of five computer vision models we acquire the following accuracy metrics:

Model Name	Accuracy Metric
Resnet152	0.108
Convnext	0.765
Cvt21	0.7804
Vit	0.518
Swin_tr	0.821

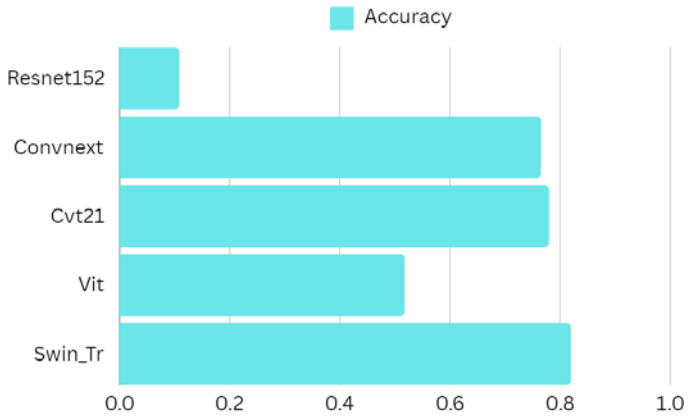
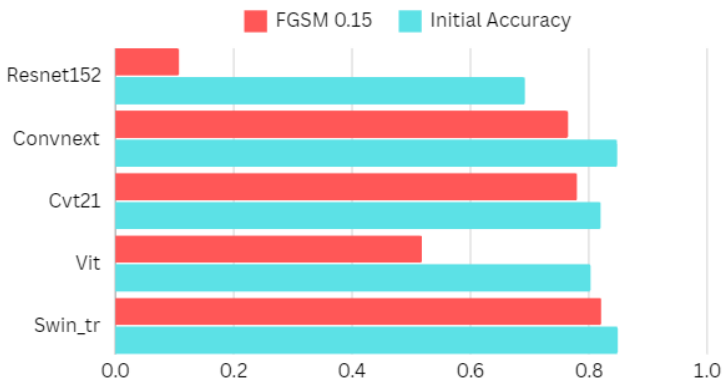


Figure 16.Accuracy Results after FGSM attack

More performance metrics without a baseline to compare will be meaningless.To assess our experiment results and come to conclusions about their robustness we will be comparing them with our baseline ImageNet experiment to be able to construct a accuracy drop table.



Model Name	Accuracy Drop
Resnet152	84.39%
Convnext	9.78%
Cvt21	3.7%
Vit	35.49%
Swin_tr	2.48%

Figure 17.Comparison of FGSM accuracy drop

Our FGSM attack robustness results don't quite mirror our previous experiment, although the accuracy drop is imperative in most of the models , on average they show a better robustness than the objectnet experiment , an outcome that can give us a paramount insight beyond the data itself, but the role of robustness in such attacks that a model architecture can provide.

5.3 Results Assesment

Resnet

The model with the highest FGSM attack success rate is ResNet-152, with an 84.39% accuracy drop. ResNet's simple architecture, compared to other models, appears to be more vulnerable to straightforward adversarial attacks like FGSM. Although its residual connection blocks greatly enhance its performance compared to other pure convolutional neural networks, they also introduce vulnerabilities. The skip connections of the residual blocks mitigate the vanishing gradient problem, but in stabilizing the gradient flow, they can amplify the calculating gradients and, indirectly, the perturbations of the FGSM attack, making the image far more difficult for the model to predict. Furthermore, the gradient flow can propagate more easily through the network due to the skip connections. This means that even small perturbations introduced by FGSM can have a larger impact on the final output compared to a network without residual connections. Another possible explanation for our results is the capacity of the residual connections to reuse features. Residual connections facilitate feature reuse across layers. Adversarial perturbations can exploit this characteristic by affecting multiple layers simultaneously, leading to more noticeable changes in the output.

Vit

The next highest accuracy drop is observed in the ViT model, with 35.49%. Although a much newer architecture than ResNet, ViT still shows susceptibility to the FGSM attack. ViT relies heavily on self-attention mechanisms to capture global and local dependencies in images. While these mechanisms can enhance the model's ability to understand complex relationships in the data, they may also introduce vulnerabilities to adversarial attacks. ViT processes images as sequences of patches, utilizing self-attention to aggregate information across these patches. Adversarial perturbations can propagate through the self-attention mechanism, creating different embedded tokens that could jeopardize the model's ability to grasp local features and, as a result, form relationships when they are used in the transformer block to extract global representation context.

ConvNext

The ConvNext model seems to be much more robust than its previous counterparts although with not an insignificant accuracy drop rate. Having a more complex design with convolutions but without the residual connections of Resnet give the ConvNext a better accuracy. Although architecturaly inspired by the transformer models, using its unique design of group convolutions seems to work better at capturing global relationships with the use of convolution than the self-attention mechanism. Also using is bottlenet block could be playing a filter advantage to clear the extra noise of the perturbations . Although still susceptible to FGSM attacks due to the nature of convolution as a global dependencies extractor from the small images patches.

Cvt21

In case of the model Cvt21 which takes the second place of the second most robust computer vision model with accuracy drop 3.7%, we can clearly see the combined benefits of the two basic model architectures the CNNs and the Transformers. Leveraging convolutional layers in the early stages of the architecture , provide a more robust foundation for the model to understand the localized features like edges and textures before feeding it to the transformer block. These blocks now can more easily capture the long-range dependencies to form the global context of the image. By combining CNNs and Transformers, CvT achieves a balance between capturing both local details and global relationships. This might make it less susceptible to FGSM attacks that primarily target localized features even if some pixels are manipulated.

Swin Transformer

For our more robust to FGSM attack computer vision model the Swin Transformer with an accuracy drop only 2.48% its success seems to be multifactored. Unlike a standard ViT which heavily relies on self-attention, Swin Transformer employs a hierarchical architecture with shifted window-based self-attention. This approach focuses on local neighborhoods within windows, offering some inherent resilience to local feature extraction . Also Swin Transformer utilizes a hierarchical structure with multiple stages. Each stage extracts features at different scales. This allows the model to capture information relevant to both fine-grained details and larger structures within the image. Even if FGSM disrupts some localized features, the model might still rely on the information from other scales for accurate classification, contributing to its robustness using these techniques as regularizers.

5.4 Evolutionary attack experimentation setting

In our exploration of robustness and vulnerabilities within computer vision models, we implemented an evolutionary attack utilizing a simple genetic algorithm on our five selected models trained on the ImageNet dataset. The genetic algorithm, inspired by principles of natural selection and genetics, was employed to iteratively evolve perturbations that could deceive the models. This attack was a targeted attack, designed with the specific objective of misleading the models into misclassifying a randomly generated image as a predetermined target class.

The evolutionary algorithm operates by generating an initial population of random perturbations, which are then evaluated based on their effectiveness in inducing the desired misclassification. Over successive generations, these perturbations undergo selection, crossover, and mutation processes to produce increasingly effective adversarial examples. The fitness function, a critical component of the algorithm, measures the degree to which the perturbed image is classified as the target class by the model. By iteratively refining the perturbations through the genetic algorithm, the attack systematically discovers the most effective adversarial examples that exploit the vulnerabilities of each model.

Implementation

Each of our model genetic algorithm attack assessment took place in a ipynb file the GeneticAlg.ipynb along with their corresponding functions.

For the implementation, we utilized tournament selection as our method for choosing individuals from the population for reproduction. Tournament selection involves randomly selecting a subset of individuals from the population, evaluating their fitness, and selecting the best individual from this subset as the winner. This method promotes a diverse set of solutions by ensuring that individuals with varying fitness levels have a chance to be selected, balancing exploration and exploitation effectively. This diversity is crucial in preventing premature convergence to suboptimal solutions and ensuring a thorough search of the perturbation space.

```
def tournament_selection(population, fitness_scores, tournament_size):
    selected_indices = np.random.choice(range(len(population)), tournament_size,
replace=False)
    selected_population = [population[i] for i in selected_indices]
    selected_fitness_scores = [fitness_scores[i] for i in selected_indices]
    winner_index = np.argmax(selected_fitness_scores)

    return selected_population[winner_index]
```

The fitness function used in our genetic algorithm was based on the model's prediction confidence for the targeted class. Specifically, the fitness score was derived from the model's probability estimate for the chosen target class.

```
def fitness_function_targeted_TRANS(model, perturbed_img,original_pred):
    tensImage=torch.Tensor(perturbed_img)
    outputs = model(tensImage)
    logits = outputs.logits
    probabilities = F.softmax(logits, dim=1)
    confidence_class_85 = probabilities[0, original_pred]
    fitness_score =confidence_class_85.item()
    return fitness_score
```

To expedite the evaluation process, we set an end condition where the algorithm terminated once the model was at least 50% confident that the perturbed image belonged to the targeted class. Upon reaching this threshold, we verified if the model indeed predicted the targeted class, thus confirming the success of the attack.

```
    if bestof > 0.5:
        max_index = fitness_scores.index(bestof)
        final_img= population[max_index]
        prediction = model.predict(final_img, verbose=0)
        perturbed_pred=prediction.argmax()
        if perturbed_pred==target:
            saveImage(final_img)
            np.savetxt('convnext.txt', final_img.reshape(-1, final_img.shape[-1]),
fmt='%.8f')
            found = True
            break
    # if population
```

For the crossover operation, we employed single-point crossover. Single-point crossover involves selecting a random crossover point along the length of the parent chromosomes and exchanging the segments after this point to create offspring. This method maintains genetic diversity while allowing for the combination of beneficial traits from both parents. Single-point crossover was chosen due to its simplicity and efficiency, which are beneficial in maintaining a

balance between exploring new perturbation combinations and exploiting existing high-fitness solutions.

```
def single_point_crossover(parent1, parent2):
    crossover_point = np.random.randint(1, parent1.size)
    child = np.zeros_like(parent1)
    flat_parent1 = parent1.flatten()
    flat_parent2 = parent2.flatten()
    flat_child = child.flatten()

    flat_child[:crossover_point] = flat_parent1[:crossover_point]
    flat_child[crossover_point:] = flat_parent2[crossover_point:]

    return flat_child.reshape(parent1.shape)
```

The mutation process in our genetic algorithm introduced random variations to the image array, encouraging further exploration of the perturbation space. The mutation was implemented by generating a perturbation matrix with values uniformly sampled from the range [-perturbation, perturbation], where perturbation defines the maximum deviation allowed for each pixel. A mutation mask was then applied, which is a binary matrix generated using a binomial distribution with a specified mutation rate. This mask determines which pixels in the image array are affected by the perturbation. The resulting mutated image is obtained by adding the element-wise product of the perturbation matrix and the mutation mask to the original image array.

```
def  mutate(img_array, mutation_rate, perturbation):
    mutation = np.random.uniform(-perturbation, perturbation, img_array.shape)
    mutation_mask = np.random.binomial(1, mutation_rate, img_array.shape)
    return img_array + mutation * mutation_mask
```

Parameter Selection

In the design and implementation of our evolutionary attack on computer vision models, careful consideration was given to the selection of hyperparameters to optimize the balance between computational efficiency and the effectiveness of the attack. These hyperparameters were selected based on a combination of empirical testing and theoretical considerations, aiming to fine-tuning the algorithm, to ensure that our evolutionary attack could effectively converge for all of our scenarios and exploit the vulnerabilities of state-of-the-art computer vision models.

The chosen hyperparameters and their respective values are as follows:

- **Tournament Size (tournament_size = 5)**: Tournament size determines the number of individuals selected randomly from the population to compete in each tournament. A tournament size of 5 was chosen to strike a balance between selection pressure and genetic diversity. A smaller tournament size can maintain diversity and prevent premature convergence, while still providing enough pressure to select high-quality individuals.

- **Population Size (population_size = 50)**: The population size specifies the number of individuals in each generation. A population size of 50 was selected to ensure sufficient diversity among the solutions while keeping the computational demands manageable.

This size allows the algorithm to explore a wide range of perturbations, increasing the likelihood of finding effective adversarial examples.

- **Number of Generations (generations = 1000)**: The number of generations defines how many iterations the genetic algorithm will run. Setting the number of generations to 1000 ensures that the algorithm has ample opportunity to evolve and refine the perturbations over time, allowing for the convergence to high-fitness solutions. This number was chosen based on preliminary experiments indicating that it provided a good trade-off between solution quality and computational time.

- **Mutation Rate (mutation_rate = 0.2)**: The mutation rate indicates the probability of each gene (or pixel) being mutated. A mutation rate of 0.2 was chosen to introduce sufficient variability into the population, enabling the algorithm to explore new areas of the perturbation space. This rate ensures that the algorithm does not get stuck in local optima and maintains genetic diversity throughout the evolutionary process.

- **Perturbation Magnitude (perturbation = 2)**: The perturbation magnitude determines the range within which pixel values can be adjusted during mutation. A perturbation value of 2 was selected to provide a balance between making significant enough changes to deceive the model along with faster convergence and enough changes to avoid local optima scenarios.

- **Target Class (target_class = 85)**: The target class specifies the class label that the model should incorrectly predict as a result of the adversarial attack. In our experiments, class 85 was chosen arbitrarily to demonstrate the targeted attack's effectiveness. This choice allows us to evaluate the model's robustness against a specific misclassification scenario.

5.5 Results

In this section, we present the results of our implemented attack and discuss their implications in the context of our robustness exploration scheme. The targeted attack was deployed on class 85 of the ImageNet dataset, which corresponds to a quail, although the attack could be adapted to target any of the other classes. The attack was successful across all the models, demonstrating their vulnerabilities. However, there is no straightforward metric to compare the performance of each model beyond analyzing the total requests made by the algorithm to achieve the defined convergence rate.

In our experiment, we define the presaid key metric as the number of generations the genetic algorithm required to make the model sufficiently confident (at least 50% probability) that the randomly generated image was indeed classified as a quail. This metric reflects the algorithm's efficiency in finding an effective adversarial perturbation. The total requests that have been made are the generation number multiplied by the population number. However, even this metric has limitations when used for comparative analysis due to the inherent randomness in genetic algorithms. Factors such as initial population conditions , complexity of the image and the stochastic nature of mutations can cause significant variations in convergence rates.

Each models generations to convergence can be seen below:

Resnet152 : After 253 Generations
Convnext : After 157 Generations

Vision Transformer : After 174 Generations
Cvt21: After 166 Generations
Swin Transformer: After 271 Generations

Figure 18.Example of a successfull classified quail from imagenet dataset

Figure 19.Convnext's prediction of a quail.

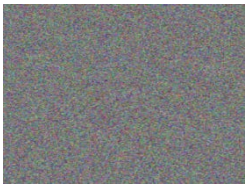


Figure 20.Convnext's Prediction of a quail.

Figure 21.Cvt21's Prediction of a quail.

Figure 22. Swin's prediction of a quail

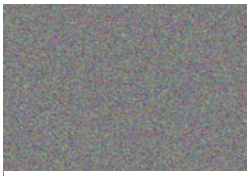


Figure 23. Vit's prediction of a quail.

5.6 Results Assessment

The results demonstrate that while all models eventually succumbed to the targeted attack, the number of generations required for convergence varied significantly.
It is important to note that our attack was conducted in a black box setting, where the internal parameters and gradients of the models were unknown. Despite this, the attack required a substantial number of requests, at least in the thousands, to achieve success. This high number of requests could potentially be detected by defensive mechanisms designed to monitor and flag anomalous patterns of access or query frequency.

To further assess the robustness of these models, we tested the transferability of the adversarial examples generated. Transferability refers to the ability of an adversarial example crafted to deceive one model to also deceive other models. In our experiments, we found that the adversarial examples did not successfully transfer to other models. This suggests that while our attack was effective in a targeted manner, its ability to generalize across different models was limited. However, this does not diminish the potential threat posed by such attacks.

Although our specific implementation required many requests and did not demonstrate transferability, there are numerous ways in which these types of attacks can be refined and implemented to pose a more serious threat . For instance using a similar model to reduce the number of queries needed, using a more optimized algorithm, or implanted them as untargeted or boundary attacks with them being harder to detect.

While current detection measures may mitigate the threat posed by high-frequency query attacks, the evolving landscape of adversarial techniques necessitates ongoing research and innovation. Ensuring the security and reliability of computer vision systems is paramount, especially as these technologies are increasingly deployed in critical applications.

In conclusion despite the non-transferability and high request count observed in our experiments, the potential for more advanced and efficient attack methodologies remains a significant concern. Future work should focus on enhancing model robustness and developing

comprehensive defense mechanisms to safeguard against the multifaceted threats posed by adversarial attacks.

6 Conclusions

In this thesis, we conduct a comparative study to explore the robustness of various computer vision models in the domain landscape. By examining distinct architectures like ResNet, CVT, ViT, ConvNext, and Swin Transformer, we aim to discern their relative robustness strengths and weaknesses. This comparative analysis provides valuable insights into how different architectural designs influence model performance under unforeseen challenges, which is crucial for their real-world deployment. We evaluated these models using the ObjectNet dataset and subjected them to adversarial attacks to assess their resilience.

In the computer vision robustness chapters we presented our models architectures and the datasets involved. After we evaluated our models performances using test samples from their training dataset Imagenet to build a baseline, we tested them on the overlapping classes of the ObjectNet dataset and we analyzed their perfomances drops in relation to the nature of data subjected and their architectures. Our findings unveiled a nuanced narrative. Despite advancements in transformer architectures like CVT and Swin transformer, none of the models proved impervious to the variations introduced by the ObjectNet dataset .

In the chapters of Vulnerabilities in Computer vision we analyzed the domain paradigm of the most known robustness related vulnerabilities and we presented a basic taxonomy of their underlying techniques. Implementing two diverse adversarial attacks using the FGSM and an evolutionary algorithm we again obtained their performance metrics and we analyzed their susceptibility regarding the nature of the implemented attacks and their architectures.

The subsequent examination under adversarial conditions, employing the FGSM attack, underscored the variability in model resilience. Surprisingly, the Swin Transformer exhibited the highest resilience, followed by CVT, ConvNext, ViT, and ResNet. This ranking challenges conventional assumptions about the superiority of newer architectures, at least in simple adversarial scenarios. Due to the adaptive nature of the genetic adversarial attack, which continuously refines perturbations to exploit model weaknesses, all the models demonstrated a significant lack of resilience . This outcome underscores the inherent challenges faced by current computer vision architectures when confronted with dynamically evolving adversarial threats, further emphasizing the necessity for innovative defensive strategies in future research.

Our exploration not only underscores the progress made in enhancing model robustness but also emphasizes the challenges that lie ahead. Robustness is not a one-size-fits-all endeavor; it is a multifaceted problem influenced by factors ranging from architecture design to dataset quality and quantity.

The journey towards robust computer vision models necessitates a holistic approach. While architectural innovations like transformers show promise, we must not overlook the pivotal role of data. Quality and diversity in training data are paramount, serving as the bedrock for building models capable of generalizing well in diverse scenarios. Moreover, the exploration of defensive mechanisms, including adversarial training and regularization techniques, remains critical in fortifying models against unforeseen adversarial manipulations. While the specifics of defense mechanisms vary, their overarching goal remains consistent: to bolster the robustness of computer vision models against adversarial threats. By integrating robustness as a core objective in model development and deployment, we can mitigate the impact of adversarial attacks and foster greater reliability and trustworthiness in AI systems.

However, it is essential to recognize that the landscape of adversarial attacks is dynamic and evolving. As such, the pursuit of robust defenses necessitates continual innovation, rigorous evaluation, and collaborative efforts within the research community. By fostering an open dialogue and sharing insights and best practices, we can collectively advance the resilience of

computer vision models and address the challenges posed by adversarial vulnerabilities.

In conclusion, our study serves as a reminder that the pursuit of robustness in computer vision models is an ongoing endeavor. While progress has been made, there is no room for complacency. As researchers and practitioners, we must continue to push the boundaries of understanding, innovation, and collaboration to navigate the complexities of real-world deployment and ensure the reliability and trustworthiness of computer vision systems.

7 References

[1] "K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.".

[2] "Mukherjee, S. (2020). The Annotated ResNet-50. Towards Data Science. Available: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758.".

[3] "Sachan, A. (2018). Detailed Guide to Understand and Implement ResNets. CV-Tricks. Available: https://cv-tricks.com/keras/understand-implement-resnets/.".

[4] "Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans,".*LA, USA, 2022, pp. 11966-11976. doi: 10.1109/CVPR52688.2022.01167.*.

[5] "Wu, H., Xiao, B., Codella, N., Liu, X., Dai, X., Yuan, L., & Zhang, L. (2021). CvT: Introducing Convolutions to Vision Transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada,". *2021, pp. 22-31. doi: 10.1109/ICCV48922.2021.00009.*.

[6] "Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)". *Montreal, QC, Canada, 2021, pp. 9992-10002. doi: 10.1109/ICCV48922.2021.00981.*.

[7] "Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words:".*Transformers for Image Recognition at Scale. In International Conference on Learning Representations (ICLR), 2021. Available: https://arxiv.org/abs/2010.11929.*.

[8] "Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F.-F. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248-255.".. *doi: 10.1109/CVPR.2009.5206848.*.

[9] "Wei, Zhipeng & Chen, Jingjing & Goldblum, Micah & Wu, Zuxuan & Goldstein, Tom & Jiang, Yu-Gang & Davis, Larry. (2023).".*Towards Transferable Adversarial Attacks on Image and Video Transformers. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society. PP. 10.1109/TIP.2023.3331582.*.

[10] "Saha, A., Subramanya, A., & Pirsiavash, H. (2020). Hidden Trigger Backdoor Attacks. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 11966-11976. doi: 10.1609/aaai.v34i01.5467.".

[11] "Biggio, B., & Roli, F. (2018). Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. Pattern Recognition, 84, 317-331. doi: 10.1016/j.patcog.2018.07.023.".

[12] "He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), Computer Vision – ECCV 2016 (Lecture Notes in Computer Science, vol 9908). Springer, Cham. doi: 10.1007/978-3-3".*doi: 10.1007/978-3-319-46493-0_38.*.

[13] "Jia, S., Ma, C., Yao, T., Yin, B., Ding, S., & Yang, X. (2022). Exploring Frequency Adversarial Attacks for Face Forgery Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022"., *pp. 4093-4102. doi: 10.1109/CVPR52688.2022.00407.*.

[14] "Arnab, A., Miksik, O., & Torr, P. (2020). On the Robustness of Semantic Segmentation Models to Adversarial Attacks. IEEE Transactions on Pattern

Analysis and Machine Intelligence, 42(12), 3040-3053. doi: 10.1109/TPAMI.2019.2927466.".

[15] "Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndic, N., Laskov, P., Giacinto, G., & Roli, F. (2013). Evasion Attacks against Machine Learning at Test Time.".*In Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), 2013, pp. 387-402. doi: 10.1007/978-3-642-40994-3_25.*.

[16] "Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. (2018). Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting.".*In Proceedings of the IEEE 31st Computer Security Foundations Symposium (CSF), 2018, pp. 268-282. doi: 10.1109/CSF.2018.00027.*.

[17] "Marchal, S., Kirichenko, A., Patel, A., Boerger, M., Tcholtchev, N., Nguyen, M.-D., La, V. H., Cavalli, A. R., Soriente, C., Kourtellis, N., Perino, D., Lutu, A., Park, S., Bagave, P., Ding, A., Westberg, M., Liyanage, M., Wang, S., Siniarski, B.".*, Sandeepa, C., & Sene, T. (2022). SPATIAL D1.2 – Security Threats Modelling for AI-based System Architectures. H2020 Project SPATIAL – Grant agreement No. 101021808.*.

[18] "Akhtar, N., Mian, A., Kardan, N., & Shah, M. (2021). Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey. IEEE Access, 9, 155161-155196. doi: 10.1109/ACCESS.2021.3127960.".

[19] "Machado, G., Silva, E., & Goldschmidt, R. (2020). Adversarial Machine Learning in Image Classification: A Survey Towards the Defender's Perspective. Available: https://arxiv.org/abs/2012.11767.".

[20] "Alhajjar, E., Maxwell, P., & Bastian, N. (2021). Adversarial Machine Learning in Network Intrusion Detection Systems. Expert Systems with Applications, 186, 115782. doi: 10.1016/j.eswa.2021.115782.".

[21] "von der Assen, J., Sharif, J., Feng, C., Bovet, G., & Stiller, B. (2022). Asset-driven Threat Modeling for AI-based Systems. In Proceedings of the 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)"., *2022, pp. 16-23. doi: 10.1109/EuroSPW56120.2022.00011.*.

[22] "Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., & Katz, B. (2019). ObjectNet: A Large-Scale Bias-Controlled Dataset for Pushing the Limits of Object Recognition Models.".*In Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 9453-9463. Available: https://arxiv.org/abs/1908.04919.*.

[23] "Sahai, A. (2021). Lecture 10: CNNs and Topics on Computer Vision. Available: https://inst.eecs.berkeley.edu/~cs182/sp21/lecture/10.".

[24] *Long, T., Gao, Q., Xu, L., & Zhou, Z. (2022). A Survey on Adversarial Attacks in Computer Vision: Taxonomy, Visualization and Future Directions. Computers & Security, 121, 102826. doi: 10.1016/j.cose.2022.102826.,* 102826. doi: 10.1016/j.cose.2022.102826..