Robust convolutional neural networks as models of primate vision

A THESIS PRESENTED
BY
BASTIEN LE LAN
TO
THE DEPARTMENT OF COMPUTER AND COMMUNICATION SCIENCES

UNDER THE SUPERVISION OF
WILL XIAO AND PROFESSOR GABRIEL KREIMAN
EVALUATED BY
JOHANNI BREA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE
IN THE SUBJECT OF
DATA SCIENCE

EPFL
LAUSANNE, SWITZERLAND
JULY 2024

Thesis advisors: Johanni Brea and Gabriel Kreiman Bastien Le Lan

Robust convolutional neural networks as models of primate vision

Abstract

Convolutional neural networks (CNNs) have revolutionized computer vision, achieving human-level performance on various tasks. However, their vulnerability to adversarial attacks raises concerns about their robustness and reliability. This thesis investigates the relationship between the adversarial robustness of CNNs and their ability to model primate vision. Building upon previous studies, we utilize a more controlled dataset and incorporate both neural monkey data and human behavioral data to assess the alignment between robust CNNs and primate visual perception. We find that robust CNNs, trained using adversarial training methods, align better with neural responses in the primate visual cortex. Our findings suggest that adversarial robustness is a crucial factor in developing CNNs that accurately model biological vision and offer insights into the neural mechanisms underlying visual perception in primates. Additionally, we find that attack images made from robust models are considered more realistic attacks for primates. Notably, one robust neural network in our study is more correlated to monkey neural data than the monkey is correlated to human data, underscoring the capability of CNNs to reach primate-level performance adversarial tasks.

Contents

Listing of figures

For my parents without whom I could never have done anything.

Acknowledgments

I would first like to thank Professor Gabriel Kreiman for giving me the opportunity to complete this research project in his lab and for his advice during this time especially pivoting quickly when the project did not work. My gratitude extends to the members of the lab. Thank you for your warm welcome and for making spending time in the lab something as interesting as fun.

I particularly want to thank Will Xiao who has been amazing, taking me on board this project, giving many advice, feedback, and time despite working on many other projects, and now giving me the opportunity to continue working on this project in the Livingstone lab. I have learned so much during this time thanks to him and I hope to continue learning lots more.

I also want to thank Johanni Brea who has through his introduction to machine learning class pushed me toward the data science Master's and accompanied me in the projects I have done in this master's with strong human values.

Last but not least I am so thankful for my friends and family, whether people I met here or back home, theses months in Boston would have not been the same without the support and love I received.

1 Introduction

Artificial neural networks (ANNs), computational models inspired by the networks of neurons in the brain, have a rich history dating back to the mid-20th century. Pioneering work by McCulloch and Pitts in the 1940s[34] laid the groundwork with their mathematical models of neurons capable of performing logical operations. This was followed by Rosenblatt's perceptron[40] in the 1950s, a model designed to mimic the learning process of neurons.

The development of ANNs has always been intertwined with advancements in neuroscience. For instance, the concept of Hebbian learning, which describes how connections between neurons strengthen with repeated activation, has been influential in shaping learning algorithms for ANNs. While early ANNs were relatively simple, the advent of deep learning in recent decades has led to the development of highly complex architectures capable of tackling intricate tasks like image recognition and natural language processing.

The study of biological vision, particularly in primates, has also played a crucial role in shaping the development of ANNs for visual tasks. Convolutional neural networks (CNNs), for example, were heavily influenced by the hierarchical structure and feature extraction mechanisms observed in the visual cortex.

In this thesis, we investigate the relationship between biological and artificial vision, exploring how insights from neuroscience can inform the development of more robust and brain-like ANNs. By examining the vulnerabilities of current models, such as their susceptibility to adversarial attacks, we aim to pave the way for future advancements in both artificial intelligence and our understanding of the brain.

1.1 The aims of this thesis

Convolutional neural networks (CNNs) have undeniably revolutionized computer vision, achieving performance equal to or even surpassing that of primates in various tasks. However, a critical vulnerability remains: CNNs are susceptible to adversarial attacks, often succumbing to subtle image perturbations that are imperceptible to humans. This susceptibility raises significant concerns about the reliability and safety of CNNs in real-world applications, particularly those where they are intended to model or even replace human vision.

Studies on large datasets have demonstrated the potential to train CNNs to be more robust against these attacks[47], even reaching a level of robustness exceeding that of primate vision[19]. Furthermore, research suggests that adversarial attacks crafted on these robust CNNs are more effective at fooling primate vision[15], implying that such robust models may better capture the nuances of biological visual processing.

However, these findings have primarily been observed in large-scale, multi-class classification tasks. The question remains: do these results hold true in more controlled settings, such as binary classification tasks? By focusing on a binary task, we can eliminate potential confounding factors associated with multi-class scenarios, such as misclassification due to the sheer number of categories or the presence of covariants. This controlled approach allows for a more rigorous investigation into the fundamental relationship between adversarial robustness and the ability of CNNs to model primate vision.

This thesis aims to reproduce and extend the findings of Gaziv et al.[15] by evaluating the adversarial robustness of various CNN training methods on a controlled, binary image classification task (human vs. monkey faces). We will investigate the alignment between robust CNNs and primate visual perception using both neural (macaque) and behavioral (human) data, examining the impact of attack direction (human-to-monkey vs. monkey-to-human) and different adversarial attack methods. Additionally, we will analyze the underlying factors contributing to the observed differences in adversarial robustness between CNNs and primate vision.

This thesis contributes to the field of computer vision by advancing our understanding of adversarial robustness in controlled settings. Specifically, by focusing on a binary image classification task, we provide a more rigorous examination of the factors influencing the susceptibility of CNNs to adversarial attacks. By integrating neural and behavioral data from both macaques and humans, we offer valuable insights into the relationship between robust CNNs and primate visual perception, potentially informing the development of more biologically plausible and resilient artificial vision models. Additionally, our findings have practical implications for the deployment of CNNs in real-world applications, particularly those where robustness to adversarial attacks is critical for safety and reliability.

1.2 The structure of this thesis

This thesis is structured as follows: First, we establish foundational concepts by reviewing relevant literature on biological and artificial vision, deep neural networks (DNNs), and adversarial attacks. We then detail our methodology, including the datasets, models, training procedures, adversarial methods, and evaluation metrics employed in our experiments. Next, we present our results, focusing on the reproduction of previous findings, the generation and evaluation of adversarial images on various CNNs, and the comparison of these results to primate neural and behavioral data. Finally, we discuss the implications of our findings, address limitations, and propose future research directions.

2 Foundational Concepts

This chapter presents a broad literature review covering the main pillars of this thesis.

2.1 Biological Vision

2.1.1 Primate Visual System Overview:

The primate visual system processes images through a series of stages, starting from the retina and ending in the inferior temporal (IT) cortex. This complex system involves many interconnected brain regions working together to process visual information.

In the retina, light is converted into electrical signals by photoreceptor cells. These signals then travel through the optic nerve to the lateral geniculate nucleus (LGN) in the thalamus. The LGN acts as a relay station, filtering visual information and integrating feedback from higher cortical areas to focus attention on specific parts of the visual field.

From the LGN, signals go to the primary visual cortex (V1), the first cortical area to get visual input. V1 extracts basic features like edges, orientations, and spatial frequencies. The information then flows through several extrastriate visual areas (V2, V3, V4, etc.), each specializing in processing different aspects of visual information, such as contours, shapes, motion, color, and form.

Finally, the visual information reaches the inferior temporal (IT) cortex, which is crucial for object recognition. IT neurons are selective for complex visual features and object categories, and their responses are thought to be the basis of our ability to recognize and categorize objects.[9,43,36]

Studies have shown that the IT cortex is central to object recognition. The hierarchical organization of the visual system, ending in the IT cortex, allows for the creation of increasingly complex and abstract representations of visual information. While early visual areas extract basic features, the IT cortex integrates these into more sophisticated representations that don't change much with viewing conditions, enabling robust object recognition.[Tanaka]

Moreover, research has shown a strong link between IT neuronal activity and behavior. For example, studies have found that stimulating the IT cortex can influence face categorization[1], and temporarily deactivating specific regions within IT can lead to specific patterns of object recognition problems[39]. These findings highlight the critical role of the IT cortex in connecting neural representations to visual perception and behavior.

2.1.2 Human vision vs Macaque vision

Studies comparing human and macaque vision have shown a lot of similarities in how their brains process visual information. Using techniques like single-unit electrophysiology and fMRI, it has been found that both species have similar retinotopic organization, orientation selectivity, and motion processing in their visual cortices[50 49 51]. They also show similar patterns in visual attention and eye movements.

But there are some important differences too. Human have more photoreceptors in their retinae, especially cones, which might give us better color vision[35]. Also, the size and organization of some visual areas, like V4 and MT, are different between the two species. This suggests they might process complex visual features like color and motion differently[37]. Behavioral studies have also shown that humans generally do better than macaques on tasks that need higher-level thinking, like recognizing objects and understanding scenes. While macaque vision is a useful model for understanding human vision, we need to keep these differences in mind when interpreting results and trying to apply them to humans.

2.1.3 Robustness of Biological Vision:

Biological vision, especially in primates, is often seen as the gold standard for robust perception. Studies have shown that primates excel at recognizing objects even when they change position, size, or viewpoint[9]. This robustness comes from the way the visual system is organized, with information being processed through multiple stages. Each stage extracts more complex features, ending up in the inferior temporal (IT) cortex, which specializes in recognizing objects. Studies have measured this robustness in different ways. They have done psychophysical experiments to see how well primates can recognize objects in different conditions, and they have recorded the responses of individual neurons in the IT cortex to different stimuli. These studies have shown that IT neurons can recognize objects under many different conditions, which allows for robust object recognition[32].

However, recent research has also found some weaknesses in biological vision. For example,[23] found that humans are more affected by spatially correlated noise than by spatially independent noise. This suggests that our visual system might have trouble processing certain types of noise. Additionally, Guo et al.[19] showed that individual neurons in the primate IT cortex can be fooled by adversarial perturbations, although they're more robust than standard DNNs. This challenges the idea that biological vision is inherently more robust than artificial vision and suggests that both biological and artificial neural networks might have similar vulnerabilities.

2.2 Biological and artificial neural networks

Artificial neural networks (ANNs) were first inspired by how the human brain works. Early work by McCulloch and Pitts proposed mathematical models of neurons that could do logical operations[34]. This idea was developed further with Rosenblatt's perceptron[40] and the backpropagation algorithm by Rumelhart, Hinton, and Williams[41]. These developments happened alongside discoveries in neuroscience, like Hebbian learning, which describes how synapses get stronger with use[21]. At first, people thought backpropagation wasn't biologically plausible, but recent research in deep learning and neuroscience suggests there might be backpropagation-like mechanisms in the brain through feedback connections[30] and cortical dendrites microcircuits[42 13].

ANNs, inspired by biological neural networks, have shown impressive abilities in various cognitive tasks. Recent advances in deep learning, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer models, have been really successful in tasks like image recognition, speech recognition, and natural language processing. These achievements are often compared to the way the brain processes information in its visual and auditory cortices[29].

2.2.1 Probing

Probing is a valuable technique used to investigate the internal representations of complex systems, such as artificial neural networks (ANNs) and the human brain. Originally employed in neuroscience to understand how the brain responds to various stimuli[6,24], this method has been successfully adapted to study artificial intelligence.

In the context of ANNs, probing involves training a simpler model to predict neural activity based on patterns within the ANN's hidden layers[2,12]. This essentially asks whether a simple model can decode the complex ”thoughts” of the ANN.

A favored approach is linear probing, valued for its simplicity and accuracy. By comparing the predictions of the linear probe to actual neural responses, particularly for novel stimuli, it is possible to assess the alignment between the ANN's representations and those of the brain. This offers insights into how well the ANN captures fundamental neural processes, a question of interest in fields such as computer vision, where understanding whether the ANN ”sees” like humans is important.

2.3 Deep Neural Networks (DNNs) for Vision

Deep neural networks (DNNs) have revolutionized computer vision, with convolutional neural networks (CNNs) becoming the standard for state-of-the-art visual recognition systems. These systems enable tasks such as image classification, object detection, semantic segmentation, and instance segmentation with unprecedented accuracy.

Inspired by the hierarchical structure of the biological visual cortex, CNNs employ convolutional layers to detect local patterns and pooling layers to downsample feature maps, effectively learning to represent visual information at multiple levels of abstraction[53,5,3].

2.3.1 DNN Architectures and Training

AlexNet[26], a pioneering CNN architecture, renewed interest in deep learning for image classification, leading to the development of deeper and more sophisticated models like VGG[45], ResNet[20], and Inception[46].

Recent advances in deep learning architectures, such as generative models (e.g., Generative Adversarial Networks (GANs))[17] and vision transformers[10], have further expanded computer vision capabilities, enabling tasks like image generation, style transfer[14], and video synthesis. Despite significant progress, challenges persist, including the need for large labeled datasets, substantial computational resources, and addressing issues of bias and ethical considerations in real-world applications.

The standard supervised training paradigm, a cornerstone of deep learning in computer vision, involves training DNNs on extensive labeled datasets, such as ImageNet. This process begins by defining a loss function to quantify the discrepancy between model predictions and ground truth labels. An optimization algorithm, typically stochastic gradient descent with momentum, is then employed to iteratively refine the model's parameters, minimizing the loss function and enhancing classification accuracy. This refinement is usually performed in batches to manage computational resources effectively. The trained model's performance is then rigorously evaluated on a separate test set to ensure generalization to unseen images, a critical factor in determining real-world applicability.

A significant advantage of working with DNNs in computer vision is the availability of open-source, pre-trained models with proven performance on large-scale datasets like ImageNet. These models have learned to extract powerful visual representations from millions of images, serving as valuable starting points for various computer vision tasks. Through transfer learning, it is possible to fine-tune these pre-trained models on smaller, task-specific datasets, often achieving state-of-the-art results with significantly reduced training time and computational resources. This democratization of access to high-performing models has accelerated research and development in computer vision and made sophisticated visual recognition capabilities more accessible to a wider audience.

2.3.2 DNNs as Models of Biological Vision

Deep neural networks have emerged as a prominent tool for modeling biological vision, particularly within the primate visual system. This approach gained traction with the work of Yamins et al.[53], who demonstrated that hierarchical models optimized for object recognition could effectively predict neural responses in the primate inferior temporal (IT) cortex. This finding sparked interest in using DNNs to understand the computational principles underlying visual processing in the brain.

Subsequent studies have further solidified the role of DNNs as models of biological vision. Khaligh-Razavi & Kriegeskorte[25] showed that deep supervised models, could explain IT cortical representations, highlighting the importance of task-specific training in shaping neural representations.

Recent advancements in this field include work by Dapello et al.[7], who demonstrated that incorporating a simulated primary visual cortex at the front of CNNs improved their robustness to image perturbations, suggesting a closer alignment with biological systems. Kubilius et al.[27] showed that shallow recurrent ANNs could achieve brain-like object recognition with high performance, further supporting the use of ANNs as models of biological vision. Additionally, Schrimpf et al.[44] introduced Brain-Score, a platform for evaluating different ANN models based on their similarity to brain responses and behavior, facilitating the development of more brain-like DNNs.

The use of DNNs to model biological vision has provided valuable insights into the neural computations and representations underlying visual perception. While challenges remain, the continued development and refinement of these models hold promise for advancing our understanding of the brain and potentially improving artificial intelligence systems.

2.4 Adversarial Attacks and Defenses

2.4.1 Adversarial Attacks

The vulnerability of deep neural networks (DNNs) to adversarial attacks has emerged as a significant area of research in recent years. Szegedy et al.[47] and Goodfellow et al.[18] were among the first to demonstrate that DNNs are susceptible to adversarial examples - images with subtle perturbations designed to cause misclassification. These early studies primarily focused on white-box attacks, where the attacker has full knowledge of the target model architecture and parameters.

The iterative Fast Gradient Sign Method (iFGSM), developed by Goodfellow et al.[18], is a technique for generating adversarial examples. This iterative process optimizes a small perturbation to the input image using gradient information to minimize or maximize an adversarial objective function. The objective function is designed to either reduce the probability assigned to the correct class (untargeted attack) or increase the probability assigned to a specific alternative class (targeted attack). This is achieved through gradient descent (for targeted attacks) or gradient ascent (for untargeted attacks), iteratively adjusting the image's pixel values to manipulate the model's output.

Subsequent research has explored various aspects of adversarial attacks, including their transferability across different models and effectiveness in real-world scenarios. Papernot et al.[38] investigated the transferability of adversarial examples, demonstrating that attacks designed for one model can often fool other models. Kurakin et al.[28] extended this work by showing the feasibility of adversarial attacks in the physical world, raising concerns about the security of DNN-based systems.

More recent studies have focused on developing more sophisticated attack methods and defenses. Carlini & Wagner[4] introduced a powerful attack algorithm capable of bypassing many existing defense mechanisms. On the defense side, Madry et al.[33] proposed adversarial training as a method to improve the robustness of DNNs by training them on adversarial examples.

2.4.2 Adversarial Training

Adversarial training has emerged as a key technique to enhance the robustness of deep neural networks (DNNs) against adversarial attacks. Madry et al.[33] introduced this approach, which involves training DNNs on a mixture of clean and adversarial examples. By exposing the model to these perturbed inputs during training, it learns to recognize and correctly classify them, thereby becoming less susceptible to future attacks. This approach has proven effective in improving the robustness of DNNs across various tasks, including image classification.

2.5 Linking Adversarial Robustness to Biological Vision

2.5.1 Adversarial Attacks on Primate Vision:

The field of adversarial attacks on primate vision is relatively new but has already yielded significant insights into the nature of visual perception in both biological and artificial systems. Early studies like Elsayed et al.[11] and Zhou & Firestone[55] provided initial evidence that adversarial images could influence humans, although these studies used perceptually salient manipulations and focused on time-limited settings.

Yuan et al.[54] made a significant advancement by developing a ”gray box attack” method to create adversarial images that could fool both monkeys and humans in visual categorization tasks. This study demonstrated that primate vision, like artificial neural networks, is susceptible to adversarial attacks, even when the perturbations are minimal and targeted.

Veerabadran et al.[52] further explored this phenomenon, showing that even subtle adversarial perturbations can reliably bias human perception in the same direction as ANNs. They also found that the effectiveness of these perturbations is influenced by the architecture of the ANN, with self-attention models being more effective than convolutional models.

These studies collectively challenge the assumption that primate vision is inherently robust to adversarial attacks and highlight the potential for subtle image manipulations to influence human perception. They also raise important questions about the similarities and differences between artificial and biological vision, and the potential implications of adversarial attacks for both fields.

2.5.2 Robustness of Biological vs. Artificial Neurons:

Guo et al.[19] directly compared the adversarial sensitivity of individual neurons in the primate inferior temporal (IT) cortex with individual units in state-of-the-art robust DNNs. They found that while IT neurons are less sensitive to adversarial perturbations than standard DNNs, they are surprisingly more sensitive than adversarially trained DNNs. Veerabadran et al.[52] further demonstrated that subtle adversarial perturbations, designed to alter the classification decisions of artificial neural networks, also influence human perception in forced-choice classification tasks. This influence is observed even with brief image exposures or extended viewing times, suggesting that both humans and machines are sensitive to subtle, higher-order statistics of natural images. Gaziv et al.[15] extended this work by showing that robustified ANNs can discover subtle image perturbations, called ”wormholes,” that can significantly disrupt human perception and categorization, challenging the prevailing assumption that human perception is robust to small image perturbations.

3

Methods

In this section, we will present the methods used in this thesis

3.1 The Dataset

3.1.1 Imagenet

The ImageNet dataset, introduced by Deng et al.[8], comprises over 14 million images hand-labeled across a vast hierarchy of 1000 categories. This dataset has been instrumental in fueling the development and evaluation of image classification and object detection algorithms. Its large-scale and diverse nature have made it a benchmark for assessing model performance and advancing the state-of-the-art in the field.

ImageNet's diverse image categories have made it the default training ground for many computer vision models. The models in use here are pre-trained on ImageNet as it gives a strong foundation for recognizing general visual features, facilitating transfer learning to the face data set that has more limited data.

3.1.2 The faces dataset

Human and monkey faces were collected from the web and from photographs taken in the lab to combine into 250 images per category. Monkey face images were excluded if they were at very oblique angles.

The original images used for generating distorted images were kept separate from the sets of 250 clean images per category used for establishing baseline accuracy. The original human faces images were obtained from the Chicago Face Database. We used 40 different images for each of four main type of manipulation method (interpolation-based, model ensemble, gray box, and non-targeted image degradation), totalling 160 original images. The original monkey faces were a set of 40 images used across all manipulation methods.

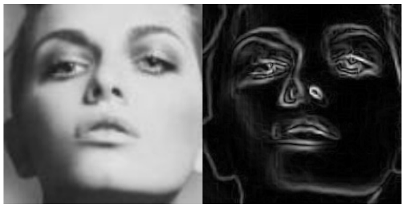


Figure 3.1: Original image and edge only image

3.1.3 Augmented face dataset: edge and style transfer

I chose to use two additional data augmentations that are later used to train different models. Making an augmented dataset came from the common issue in CNNs[16] that they are overly sensitive to texture and this might have impact the creation of the adversarial images. To solve this issue I followed two paths, one recommended in[16] making the network completely agnostic to texture, and another keeping the original images but adding edge detected images.

- Edge augmentation: We passed the image through a Sobel filter that allows to compute a local gradient and extract where the images are changing the most. This creates an edge detection function and then normalizing the image to a 0-255 range allows to create an image showing only the edges of the original one;

- Style transfer augmentation: We utilized a style transfer method developed by Huang and Belongie[22] to apply artistic styles to photographs. This technique allows us to transform images into ones mimicking the style of different artists and paintings. This approach can apply any style to any image. The approach uses a pre-trained encoder (typically based on VGG-19) to extract feature maps from both content and style images. The encoder analyzes both the content image (the monkey and human pictures) and the style image (the artwork whose style we want to copy).

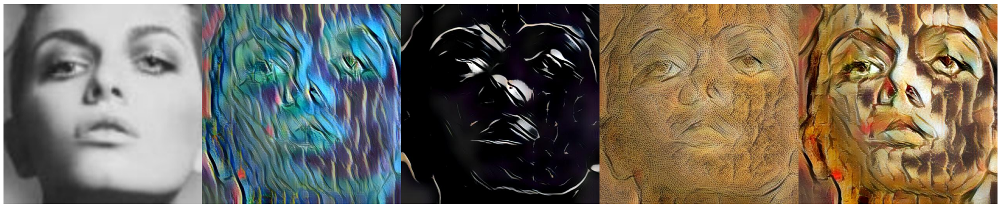


Figure 3.2: Original image side to side with different style transfers on to the same image

It breaks down these images into essential features and then aligns the mean and variance of the content features to match those of the style features. This operation transfers the style information while preserving the content structure. Finally, the decoder takes this combined information and reconstructs it into a new image that has the content of the original photograph but the artistic style of the chosen artwork. This entire process happens in one forward step, making it possible to experiment with many different styles in real time. The GitHub repo `https://github.com/naoto0804/pytorch-AdaIN.git` is also impressively well-coded and allows for an out-of-the-box execution by following simple steps in the readme;

3.2 Statistic tools

3.2.1 SVM

Support Vector Machines (SVMs) are a type of supervised machine learning algorithm used for classification and regression tasks. They work by finding an optimal boundary (a hyperplane in high-dimensional space) that best separates data points into different classes. This boundary is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points of each class. These nearest points are called support vectors, as they are the most critical data points in determining the position of the hyperplane. SVMs can handle both linear and non-linear data by using kernel functions, which transform the data into a higher-dimensional space where a linear separation might be possible. SVMs are known for their effectiveness in high-dimensional spaces and with limited data samples.

3.2.2 PCA

Principal Component Analysis (PCA) is a dimensionality reduction technique that simplifies complex datasets while preserving essential information. It works by identifying the directions, known as principal components, in which the data varies the most. These principal components are orthogonal to each other, ensuring that they capture distinct aspects of the data. The first principal component explains the largest amount of variance in the data, the second principal component explains the second-largest amount of variance, and so on. By transforming the original variables into these new, uncorrelated principal components, PCA reduces the dimensionality of the data while retaining the most important information. This makes it easier to visualize, analyze, and interpret the data, particularly in high-dimensional datasets where patterns and relationships may be difficult to discern.

3.2.3 Pearson's correlation

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2} \tag{3.1}$$

Pearson's correlation, also known as Pearson's r, measures the strength and direction of a linear relationship between two continuous variables. To compute it, we first calculate the covariance, which assesses how much the variables change together. The covariance is then normalized by dividing it by the product of the standard deviations of both variables. Standard deviation quantifies the amount of variation or dispersion of a set of values. The resulting Pearson's correlation coefficient ranges from -1 to 1. A value of -1 indicates a perfect negative linear relationship, 0 indicates no linear relationship, and 1 indicates a perfect positive linear relationship. The absolute value of the coefficient represents the strength of the correlation, while the sign indicates the direction.

3.3 Models

3.3.1 Resnet50

ResNet50, a deep convolutional neural network architecture, was employed in this study due to its proven efficacy in image classification tasks. This 50-layer network, introduced by He et al.[20], utilizes skip connections to address the vanishing gradient problem common in very deep networks. These skip connections, also known as residual connections, allow the network to learn residual functions with reference to the layer inputs, facilitating the training of deeper networks. ResNet50's architecture consists of five stages, each comprising multiple convolutional layers followed by batch normalization and ReLU activation functions. The network's depth allows for hierarchical feature extraction, potentially capturing both low-level features (e.g., edges and textures) and high-level, abstract representations relevant to our image analysis tasks. We used a pre-trained version of ResNet50 on the ImageNet dataset and fine-tuned it on our specific face data to leverage transfer learning, thereby enhancing the model's performance on our specialized task while reducing training time and data requirements.

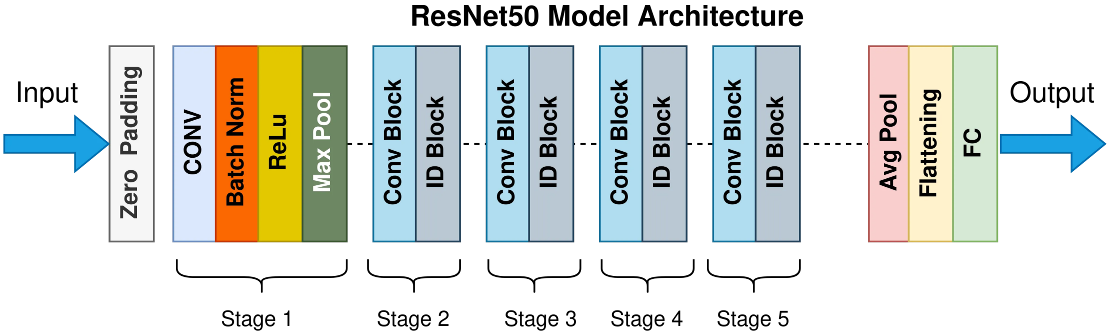


Figure 3.3: ResNet50 model schematic

3.3.2 ResNet50 with modified input and output layers

The ResNet50 being trained on imagenet had as input 3 channels images and outputs a number between one and 1000 as the category of the image. However, our face dataset is in grayscale and we want the result of the prediction to be only a human face or monkey face and not a number between 0 and 1000. To address these issues we added an input layer with a convolution filter of 1x1x3 to keep the image fixed but simply push it into three channels. This filter was initialized at 1 to keep the image intact just across the three channels, however it was free to learn during the fine-tuning.

For the output layer modification, we started by running the training images through the network without the last layer, to get a higher dimensional response that has already extracted all the interesting features of the images and that should be capable of classifying any imagenet image into the relevant category. Using this data, we fitted an SVM to try to separate the two categories from the output of the layer before the last. This method gave very good results for the SVM, and we, therefore, took the parameters of the separating hyperplane to use as parameters for the last layer of our network. These modifications allowed us to transform a pre-trained Resnet50 with performant feature extraction weights into a model specifically made for classifying our grayscale images into the two categories.

3.4 Training

3.4.1 Resnet fine tuning

Despite the last layer of the network being built from an SVM hyperplane splitting the face dataset into two 'perfect' splits, we observed that fine-tuning the model on the same training set that was used to create the hyperplane yielded better results. This can be explained by the way the SVM is made, only depending on the support vector points and not using the weight of the other points. We therefore fine-tuned our models using the training set for 50 epochs, 20 batch size, and 0.01 learning rate this allowed us to reach a validation set performances of 100% accuracy on the vanilla neural network.

3.4.2 Adversarial training and model robustness

With the goal in mind to reproduce and build upon Gaziv et al.'s[15] results on adversarially robust neural networks being better models for the vision, one critical method is the training of neural networks with the desired robustness level. To achieve this we used two different methods, both using the robustness library maintained by the Madrylab. The first method is to take a normal pre-trained ResNet50, apply the steps described above for fine-tuning to get a functional network, and then apply the adversarial training from the robustness library to fine-tune our model. The other option is to start from an already robustly pretrained ResNet50 such as those from the Madrylab's hugging-face's repo. In this case, all the weights of the network are already trained to be adversarial robust to the level we wish, and we can prepare it in a similar way to the original neural network, needing only a few fine-tuning steps. This second method also gives us reassurance in the fact that the models have been trained enough to reach the desired robustness level. Due to having 15 models to train from the 3 different training sets, and 5 different robustness levels we chose, the already robustly pre-trained neural nets allowed for faster training (still keeping the robust training during fine-tuning) and was what we used.

To enhance the resilience of our ResNet50 model against adversarial attacks, we employed adversarial training, a technique where the model is exposed to intentionally perturbed images during the learning process. We utilized the Projected Gradient Descent (PGD) algorithm to generate these adversarial examples. PGD is an iterative optimization process where, starting from a clean image, small perturbations are gradually introduced in the direction that maximally increases the model's error. These perturbations are constrained within a defined limit (ε) to ensure they remain subtle. The process is repeated for several steps, with each step involving a calculation of the loss gradient (the direction of greatest error increase) and then a projection back to within the allowed perturbation limit. By training the model on both clean and adversarial examples, we aimed to force it to learn features that are robust to (PGD-based) adversarial perturbations, ultimately improving its overall resistance to adversarial attacks. The epsilon parameter is the one we refer to as the robustness level of the network, as it is the limit until which the model is trained and therefore should be robust to attacks up to this point.

One might also wonder why not choose all the time a very large robustness level to be able to resist any attacks.

This can be quickly answered by the 'no free lunch' rule: The more robust a model is, the less accurate it gets. As an example, we can use a toy idea of having unlimited pixel budget to attack the human face images, and we can therefore transform it completely into a macaque face. A infinitely robust model would therefore be trained to classify macaque faces into humans and vice versa, making the model predictions completely wrong. Therefore at epsilon = 5, since the model only has 50% of correct predictions on imagenet, we do not look into more robust models.

3.5 Adversarial methods

- Linear Interpolations:

  $$x_{\text{adv}} = (1 - \lambda)x + \lambda x_{\text{target}}, \tag{3.2}$$

  We used linear interpolation to create a smooth transition between two images. This method blends pixel values, creating a series of morphing images. We chose the goal image by measuring distances between images in each set and selecting the closest ones.

- Aligned Linear Interpolations:

  Aligned linear interpolations (ALI) improves on standard linear interpolation by adding an alignment step. This helps preserve image structure and semantics better. Unlike regular interpolation, ALI tries to match features or regions between input images before interpolating. We again measured distances between aligned images and picked the smallest one for our attack images.

- Cycle Generative Adversarial Networks (GANs):

CycleGAN is a deep learning model that excels at translating images between two distinct categories without relying on paired training examples. It achieves this using two generator networks, each responsible for converting images in one direction, and two discriminator networks, which evaluate the authenticity of the generated images.

The model's core innovation lies in its cycle consistency loss, which enforces a constraint that the translation process should be reversible. This means an image translated from category A to B, and then back to A, should closely resemble the original image. This clever mechanism, combined with adversarial training from the generator versus the discriminator of a traditional GAN, enables CycleGAN to learn complex mappings and produce high-quality image translations. In the case here, the two categories are human faces and monkey faces however, to respect the pixel budget constraints, a linear interpolation from the original to the new category image is done.

- Model Ensemble:

  Adversarial images were created for a group of convolutional neural networks (CNNs). ImageNet pre-trained CNNs were fine-tuned on human and monkey face categories, making an ensemble of 14 fine-tuned CNNs including Inception, ResNet19, ResNeXt20, DenseNet21, and SENet22. Adversarial images were made for this ensemble using iterative gradient descent.

- Gray-box Methods:

  These methods were based on a substitute model of macaque visual neuron responses.

  A fine-tuned ResNet-101 from the ensemble above was used as the model of primate vision. A linear mapping was fitted from the last convolutional layer to the responses of neurons, using a dataset with about 1,000 pictures of faces and other objects. The model captured around 40% of response variance on held-out images. The modeled neurons were recorded in the same chronic recording array for neurons used in later experiments, but the model was fixed for making images before testing any of them. This neuron-fitted CNN acted as a model for the primate visual system.

To create adversarial attacks, either two or four different methods were used, depending on the attack direction. Two objective functions were tested: iterative gradient descent with or without l2-projection (l2-PGD). The l2-PGD projects to a fixed noise level at each step of gradient descent. This was applied in both directions on all neurons modeled by the network and is called the pattern method. The exploration of the results showed that one neuron seemed to respond to monkey features. So, another method called single neuron was created, which tried to optimize a human image for this specific neuron to make it fire to monkey faces.

3.6 Evaluation of images

3.6.1 Subjects

Three adult male monkeys were included in this study: two macaca mulatta (9–12 kg; 5–13 years old) and one macaca nemestrina (13 kg, 11 years old). The monkeys were kept in standard quad cages with 12/12 hr light/dark cycles. Before training, custom-made titanium headposts were implanted on the monkeys. After several weeks of fixation training, a second surgery was performed to implant arrays.

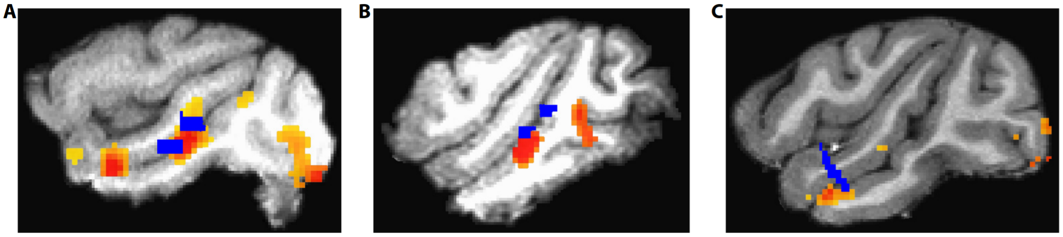


Figure 3.4: Image of the MRI of the probes

Monkeys P, R, and B1 were implanted with chronic microelectrode arrays (MicroProbes, Gaithersburg, MD). Recording sites were targeted to the medial-lateral (ML) face patch for monkeys P and R and the anterior medial (AM) face patch for monkey B1. The array targets were found using fMRI aligned to CT scans, and during surgery, landmarks from the CT scans were used. After surgery, the locations were confirmed with a second CT scan.

Extracellular electrical signals were recorded using the Omniplex data acquisition system (Plexon, Dallas, TX). All surgeries were performed under general anesthesia and sterile technique. The procedures on non-human primates were approved by the Harvard Medical School Institutional Animal Care and Use Committee and followed NIH guidelines from the Guide for the Care and Use of Laboratory Animals.

Human behavioral experiments were conducted online on Amazon Mechanical Turk. All participants provided informed consent and received monetary compensation for participation in the experiments. The experiments were conducted according to protocols approved by the Institutional Review Board at Boston Children's Hospital.

3.6.2 Quantification of image change

The amount of image change (noise level) was quantified in some cases as Mean Squared Error and in some others as the L2 norm. These are explained by a different measure of noise used in[15] that we used as a sanity check and the[54] manuscript that chose 10 MSE and our use of this data later on.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i,j}(x'_{i,j} - x_{i,j})^2, \tag{3.3}$$

$$l_2 = \sqrt{\sum_{i,j}\left(\frac{1}{255}(x'_{i,j} - x_{i,j})\right)^2} = \sqrt{\frac{N}{255^2}\cdot \mathrm{MSE}} \approx 0.8784\sqrt{\mathrm{MSE}}. \tag{3.4}$$

x′ and x are a pair of images, and i, j index N total pixels. All images were standardized to a size of N = 224 × 224 pixels, with one-channel (gray scale) and pixel values in the range 0–255. When the image size is fixed, the MSE is related to the l2-norm of an image change with pixel values in the range 0–1 3.4.

3.7 Neural evaluation of images in macaques

3.7.1 Neuron-level experiments

We recorded how neurons responded when we showed images to the monkeys during a passive fixation task. We displayed the images on an LCD monitor (ASUS VG248, 165 Hz) placed 57–61 cm from the monkeys. The monkeys fixated within a 1.5–3 degree window in exchange for juice rewards. We tracked their eye position using an infrared eye tracker (EyeLink, Ottawa, Canada). The images were 4 degrees of visual angle in size and we showed them in random order in the neurons' receptive fields. Each image appeared for 100 ms, with a 150 ms gap between images for monkey 1 and 400 ms for monkey 2 because recorded neurons had slower responses. We used an analog photodiode signal to align image onset times with the neural data.

We calculated neuronal responses as firing rates within a specific time window after the image appeared, averaged over trials. We chose this response window for each session to maximize split-half self-consistency (how consistently a neuron responds to the same image). For monkey 1, the response windows started between 60–90 ms and ended between 240–280 ms. For monkey 2, they started between 125−180 ms and ended between 340–400 ms. We only used visually selective units, which we defined as those with a split-half self-consistency higher than 0.1. This 0.1 cutoff is generous based on previous experiences and previous work. We got similar results when we tried different selection criteria. Before further analysis, we standardized each neuron's responses to have zero mean and unit variance across images.

3.7.2 Quantifying neuron deception success

Neuron-level deception success was quantified using linear Support Vector Machines (SVMs) as implemented in the Python package ”scikit-learn”. SVMs were trained separately for each experimental session. The SVMs were trained to categorize clean images (two-way categorization) using the corresponding neuronal responses. For example, in the monkey→human perceptual change direction, the SVMs were trained to classify each image as either a monkey face or a human face. We trained SVMs with balanced samples for 250 train-validation splits, leaving two images out (one from each class) in turn in each split. We trained both linear and radial-basis function (RBF) SVMs, resulting in a total of 500 classifiers per experimental session. The trained SVM was used to classify the held-out clean images, distorted images, and control images.

We measured perceptual shift success for each image as the fraction of classifiers that classified it as the target class. This value was between 0–1 but usually close to either 0 or 1. We further averaged this value across sessions and monkeys.

We also used the distance to the SVM hyperplane to measure neuron deception. This measure averaged across sessions and monkeys, let us quantify more precisely how far inside or outside the target category the attack images were. We scaled and shifted this number so that -1 represented the original category and 1 represented the target category.

3.8 Behavioral images assessment by humans through Mechanical Turk

3.8.1 Behavior-level experiments.

Subjects on Amazon Mechanical Turk were invited to perform an image categorization task. They were instructed to determine ”whether each image is a human face or a monkey face”. Subjects were instructed to ”Make your best guess. Sometimes, it may be hard to determine the correct answer.” To answer, subjects pressed the left arrow key for ”Human face” and the right arrow key for ”Monkey face”. Each image was presented for 1.5 s. There was no time limit for a response. No feedback was provided. Distorted images were randomly and evenly intermixed with clean images, which allowed us to monitor performance. We selected subjects with > 95% accuracy on clean images to include in further analyses. Responses were collected from 4–5 subjects per image.

3.8.2 Quantifying behavior deception success

Behavior-level deception success was quantified as the fraction of subjects that chose the target category for each image (each subject only saw each image once). Success rates for a method at a given noise level were calculated as the average success over images.

4

Results

4.1 Reproduction of Gaziv et al. Results:

Replicating Gaziv et al.'s findings on the vulnerability of image classification models and the effectiveness of adversarial training allowed to validate my models and methods. Previous research demonstrated the vulnerability of standard image classification models (like the epsilon = 0 ResNet-50) to adversarial perturbations,

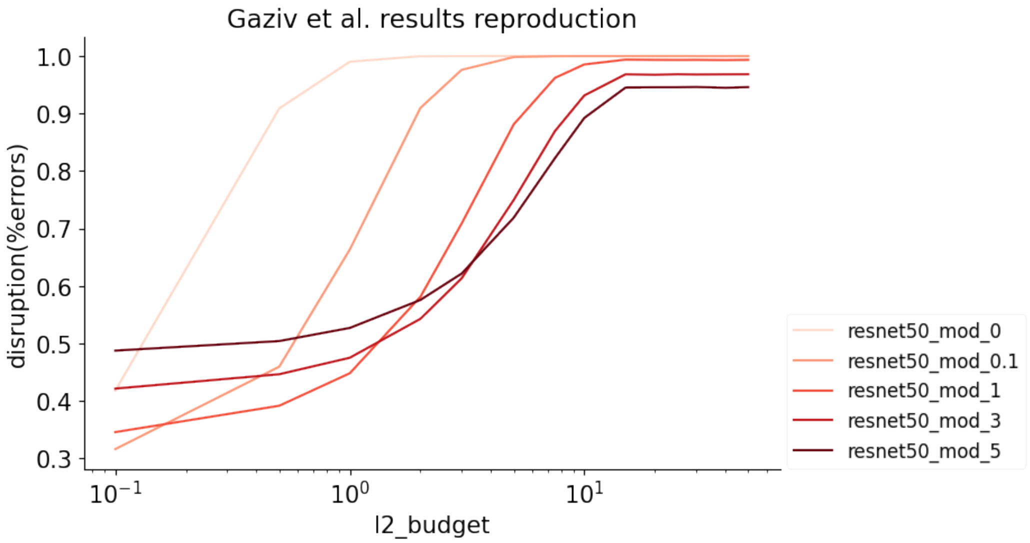


Figure 4.1: Misclassification of ImageNet categories in function of the adversarial budget (l2 budget) showing across different robust models (zero being a non-robust model and five being the most robust model in our case)

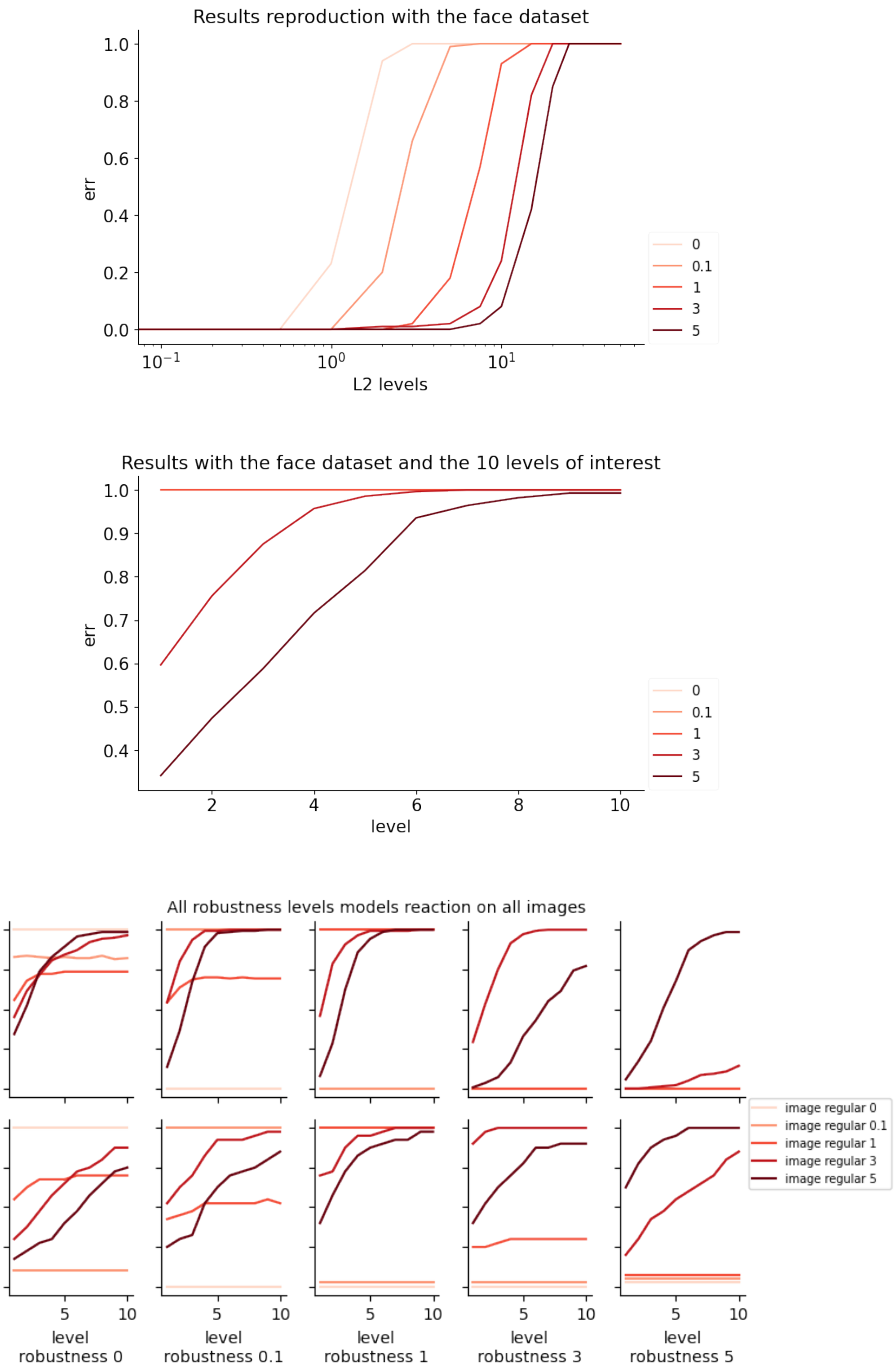
highlighting the need for robustly trained models. The replication of results reaffirms the susceptibility of unmodified models to minor image alterations and validates the effectiveness of adversarial training in improving model resilience. Notably, the results illustrate that models trained with higher epsilon values exhibit increased resistance to perturbations, confirming the efficiency of this approach in enhancing robustness. This replication establishes a solid foundation for my thesis using validated methods and models having consistent behavior under perturbation performing as expected.

4.2 Generation of Adversarial Images:

Having validated methods, we moved to the face dataset to see if we could demonstrate similar robustness of adversarially trained models on the binary classification task rather than the 1000-way classification on imagenet. In a binary classification, the model has to classify the attack image as the other class whereas it can be any of the other 999 classes in the imagenet case. Moreover, the face images are centered and scaled to take the full image space whereas categories in imagenet might only take a small portion of the image. These two factors make the creation of adversarial attacks from the face dataset more difficult.

We can see in Fig 4.2 that the trends observed by Gaziv et al are also present in our case, where the models do not get fooled as easily at the lower levels but with higher pixel budgets the models get fully fooled. One important thing to notice here is the switch from l2 noise levels ranging from 1 to 50 in the reproduction of[15] to 10 noise levels ranging from 12.4 to 24.8 (MSE 200 to 800). This range is explained by a different measure of noise used in the[54] manuscript that chose 10 MSE rather than the l2 norm and our use of this data later on. The two being linked monotonically the choice of one or the other does not make any difference. On the other hand, the choice of the range has a large impact, as a budget too small will not manage to fool the network and one too large can fool it by simply changing all the pixels. Moreover, budget steps being too large can cause a loss of information in the transition region between zero to full adversarial attack success.

The figure panel C illustrates in more detail the attack success rates by showing the performance of all robust models against images created from all models. This is also shown depending on the attack direction human to monkey (h2m) and monkey to human (m2h). We can see that the more robust a model is the less it gets fooled by the attackers from the non-robust models. The more robust models are only sensitive to their own attack or the ones from the higher robustness. However, models tend to get more fooled by attacks from themselves than from more robust models. We can moreover observe that in the h2m at low noise level, the attacks have a lower misclassification rate than in the m2h attacks.



(a)

(b)

(c)

Figure 4.2: Misclassification of human and monkey faces in function of the adversarial budget (l2 budget) showing across different robust models (zero being a non-robust model and five being the most robust model). A shows very similar results compared to Gaziv et al., while figure B and C are more zoomed in to the results of interest, and the attacks across the different models.

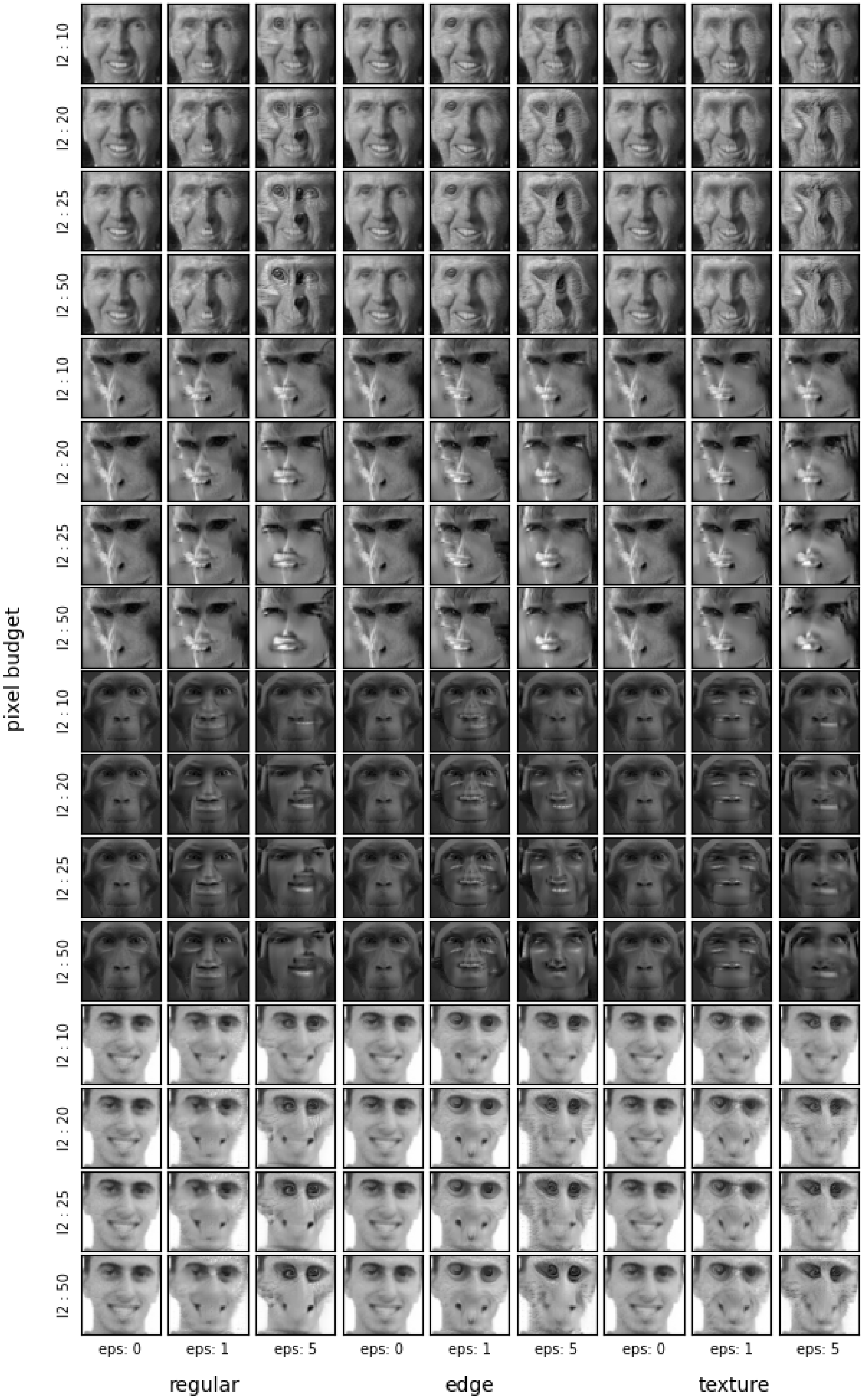


Figure 4.3: Example on 4 different images of attack using different noise budgets, robustness level, and training sets

Fig 4.3 shows examples of attacks on 4 images using 4 noise levels (as the lines), 3 robustness levels (in the columns groups), and 3 different training data (groups of 3 columns). The more robust a model is, the more effective the attack feels to our vision. Furthermore, as we may expect, the more pixel budget an image is allowed, the more realistic the images look.

4.3 Preliminary Behavioral and Neural Results on Robust-Model-Generated Images:

I created adversarial images for robustified models. Will Xiao helped a pilot experiment to show attack images, mixed with some of the clean training images, to a macaque and collected neural responses to these images. Using the training images, we fit an SVM to separate human faces from monkey faces with a success rate of 95%. Using this SVM we projected the attack images and also quantified the the success rate of these attacks under binary classification.

We can see in Fig 4.4 that the two attack directions differ greatly in the success rate of targeted attacks in each direction. Indeed in the h2m direction, the robustness level of the image generation model barely has any effect on the classification mean error which never reaches above 5%. On the other hand, in the m2h direction, more robust models generated more effective adversarial images for monkey neurons up to epsilon = 1; the images reached 75% of efficiency in fooling the monkey at epsilon = 5.

We moreover tested two different types of Resnet50, one normal, and one wide resnet50. The main differences between those two are the width of the channels and therefore, the training time. However, we can see 4.4 that the wide ResNet does not bring advantages, and it was therefore investigated no further for the rest of the thesis.

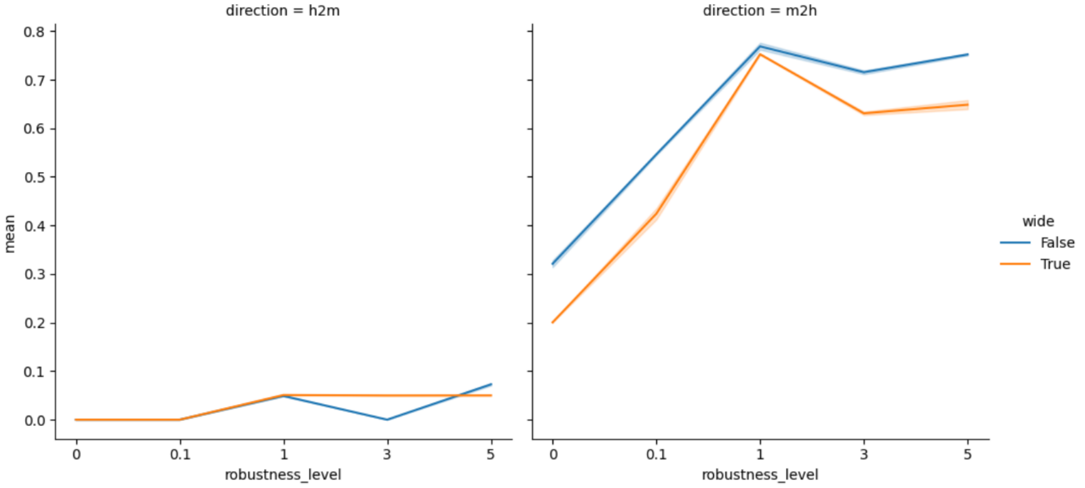


Figure 4.4: Mean percentage change in macaque neural data as a function of the robustness level of images at noise budget 7: We see the misclassification rate in h2m direction and m2h direction of both regular and wide Resnet50 used to create attack images. Notably, a large difference in misclassification can be observed between the h2m and m2h categories

4.4 Investigation of the attack efficiency imbalance between the directions

A big question throughout the study has been why is there such a difference in the efficiency of the attacks depending on the direction. This first pilot study in section 4.4 clearly shows this imbalance with the attack images barely reaching a 5% success rate in the case of h2m and 75% in the m2h case. I investigated different hypotheses to try to explain this throughout the thesis.

The first one was that the model was only learning the texture of the faces and not the shapes from it as was suggested by [16]. I therefore created three new model training categories:

- edge training:

  Adding images preprocessed by edge detection, to encourage the model to learn the

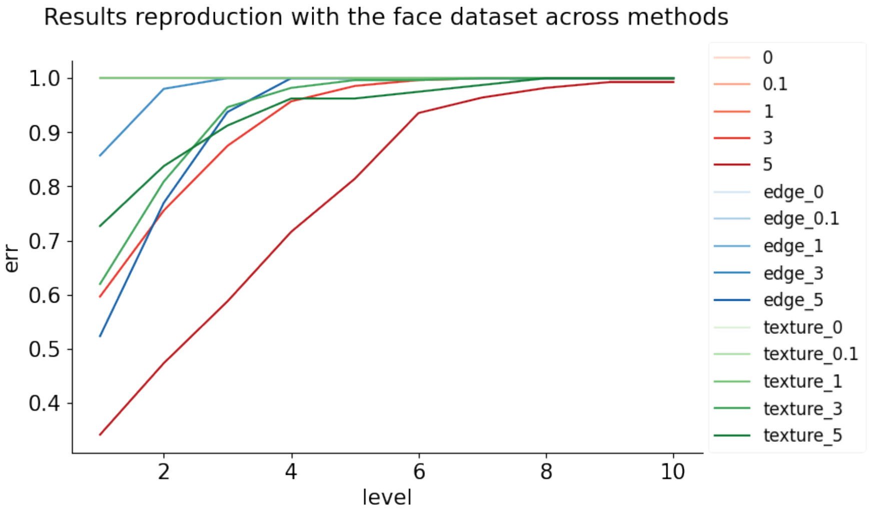


Figure 4.5: Misclassification of human and monkey faces in function of the adversarial budget (l2 budget) showing across different robust models (zero being a non-robust model and five being the most robust model in our case) and different training sets (edge augmentation and texture augmentation). Only models with robustness 3 or 5 are not fully fooled and do not disappear in the horizontal line at misclassification 1.

shapes while keeping the texture knowledge.

- texture agnostism:

  Using style transfer to augment the data and become completely texture-agnostic and only rely on the shapes.

- Random croping:

  The images were randomly cropped to try to encourage the model to learn individual parts of the face.

I quickly rejected the third option as it did not make good predictions, and became even more dependent on texture.

The figure4.5 shows the same results as Fig 4.2B shown across our two new ways of training the model now. We can see that only noise levels 3 and 5 are not fully fooled by the noise levels we are interested and this is the case in the regular but also edge training and texture agnostic. All the other robustness levels overlap the horizontal line at err = 1. We conclude that regular training is more robust than the other two training sets, as it gets less fooled by the attacks at lower levels.

However, this plot only shows the results of one model on its own attack images, so it is interesting to compare how the models do against other adversarial attacks.

In Fig 4.6, we have a complete plot of how the models get fooled by images created under different conditions, the red lines representing the texture-agnostic adversarial images and the blue ones the edges images.

We see the efficiency in fooling 15 differently fine-tuned Resnet50 depending on the pixel budget allowed for the adversarial image. The different training sets with the augmentations are grouped by columns of 3 and the 5 different robustness levels can be read across the rows. There are then the 2 attack directions, represented by the groups of 3 columns, the first 3 being the h2m direction and the last 3 being the m2h direction.

We can observe with no surprise that the more robust a model is the more difficult it becomes to fool. Surprisingly, we observe in the most robust models (the last row) that the regular training is partly robust to both attacks at low levels, the edge training gets easily fooled by the edge model images created with high robustness and the same goes for the texture models.

We can see that the strong edge training attack is also fooling the texture agnostic models in the h2m direction, whereas the strong texture attack images manage to fool the edge model in the m2h context.

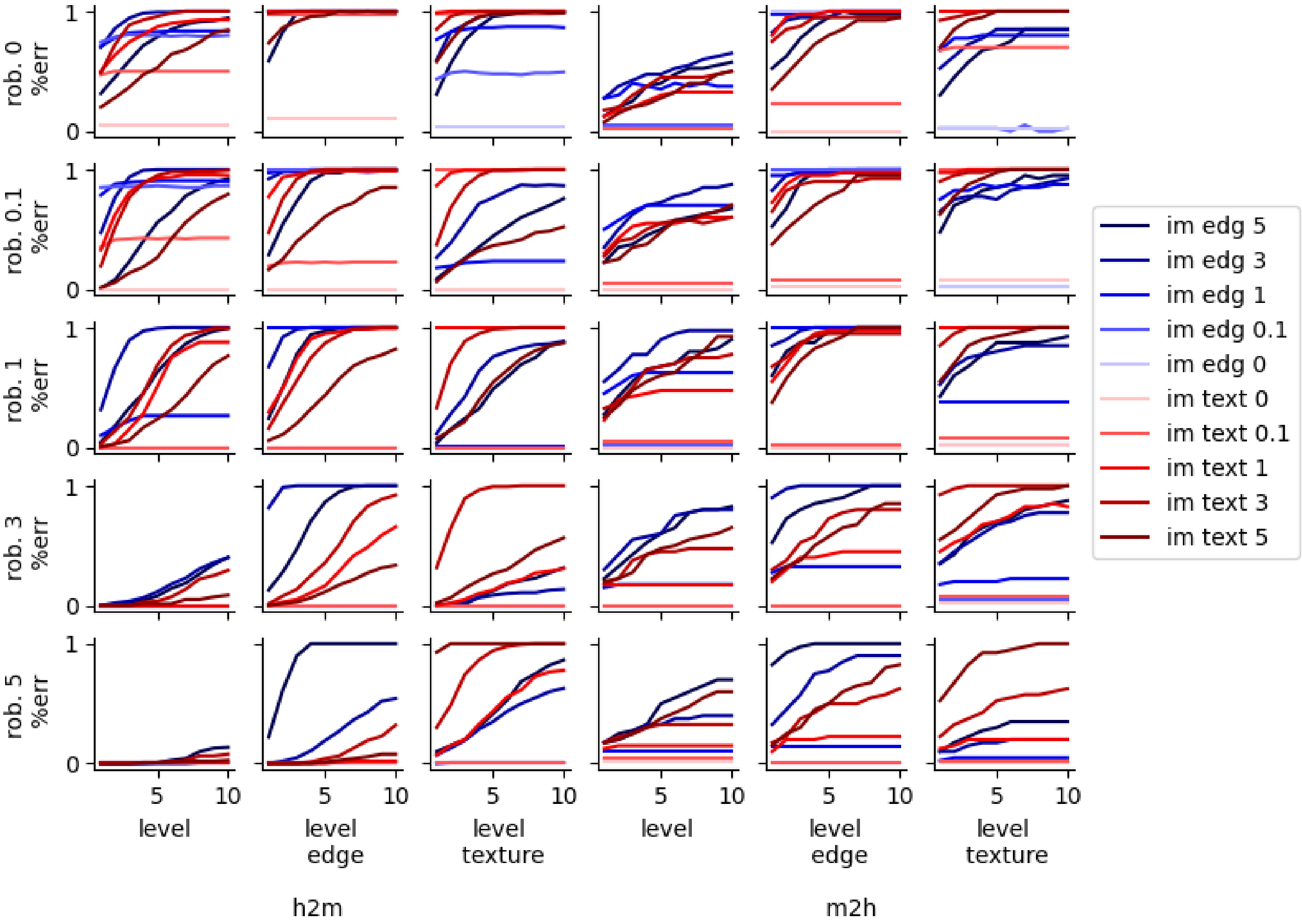


Figure 4.6: In-depth observations of misclassification rates per model in function of the noise budget and attacks that are no longer only made from the model itself. The rows represent different robustness levels of models, the columns grouped by 3 represent the attack direction, h2m in the first 3 and m2h in the last 3, and inside these groups, there are 3 different fine-tuning, regular, edge data augmentation and texture agnosticism. Each color line in the plots represents attack images created by one particular model, in red are the texture, and in blue the edge models, and the gradient of color represents the robustness of the model, the stronger the robustness the darker the color.

Comparing the regular models (columns 1 and 4), we can observe in the first rows of models stronger robustness in the m2h direction, however, this trend inverses in the most robust models (the lower rows) where the m2h direction is more susceptible to adversarial attacks.

A similar trend however to a lesser extent is also seen in the edge columns (2 and 5) where the first row h2m is more susceptible to attacks then m2h but the lower rows follow an opposite trend.

In the case of texture, it is not so clear, a similar trend seems to emerge but the last row goes against it with a stronger adversarial sensitivity in the h2m direction.

One hypothesis to explain this trend could be that adversarial training is effective in the h2m direction which is in the non-robust case more susceptible to attacks. The adversarial training would focus on solving these particular attacks, not touching the m2h attacks.

The two model augmentations gave satisfactory results and images that seemed more realistic to me. We therefore conducted a second pilot experiment using these attack images along with the the regular model's attack images on the same macaque studied previously. Fig4.7 shows these results, indicating no obvious effects from the data augmentations used. However, the success rate, which measures the mean of (binary) success or failure in fooling the monkey, may obscure subtle effects of the data augmentation. Therefore, I further analyzed the mean distance to the SVM hyperplane, a graded measure. From this, we can see that the training set still doesn't bring any particular advantages, we however see that at low robustness the m2h attacks are already closer to the SVM hyperplane. Similarly to our first pilot study, the robustness changes the efficiency of the model in the m2h direction but not in the h2m one.

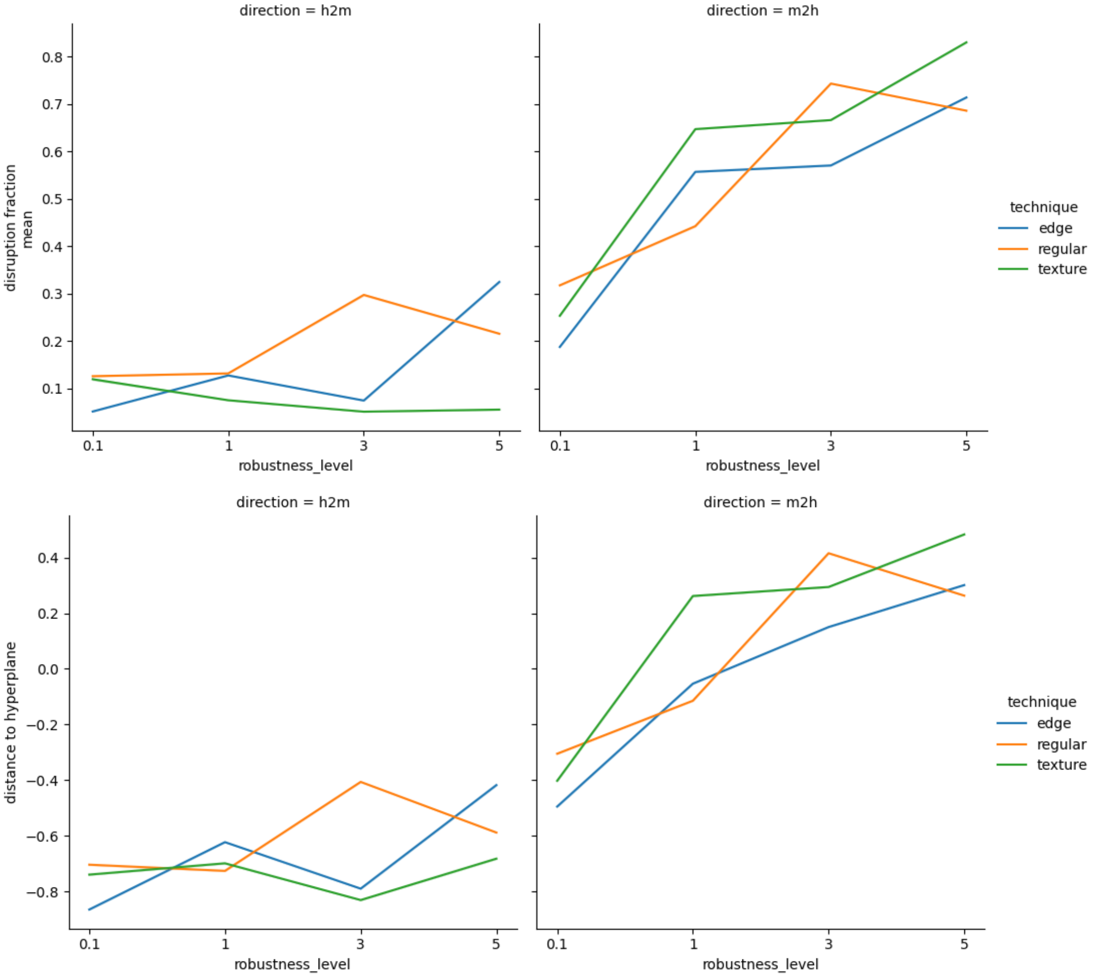


Figure 4.7: Disruption level and distance to the hyperplane in function of the robustness level and the training set The disruption fraction being the fraction of points crossing the hyperplane, it is interesting to look into how close or far the points are to the hyperplane.

4.5 Comparison to Yuan et al. Adversarial Images:

Using our different models, it is possible to evaluate them on previously created images by[54]. This evaluation is not yet possible on the images created by the new models as we only have pilot data that contains only one noise level and no human psychophysics measurements.

These images created with a broad range of methods and different techniques have one obvious advantage on the images I created as they have already been evaluated on Mechanical Turk and have corresponding monkey neural responses. It is therefore possible to use them to compare our model to aspects of vision in both primate species. Using the same figure organization as 4.6, we observe in fig 4.8 for the regular model and the edge data augmentation in the h2m direction that the more robust a model is the less susceptible it is to adversarial attacks. In the texture agnostic model, we see that the epsilon = 0 model is only fooled to some extent by the attacks, however, every robustness level except for the stronger one barely gets fooled by the images and keeps close to 0, as if the style transfer training had rendered the model way more robust. Surprisingly, the most robust network gets fooled by the cycle GAN interpolation, and the linear and aligned linear interpolations. In the m2h direction, all three training sets perform similarly at the same robustness, with linear, aligned linear, and GAN interpolations being effective in fooling the network on some of the images and the other methods not working. The effect of robustness is small but the more robust network resists a little better the adversarial images.

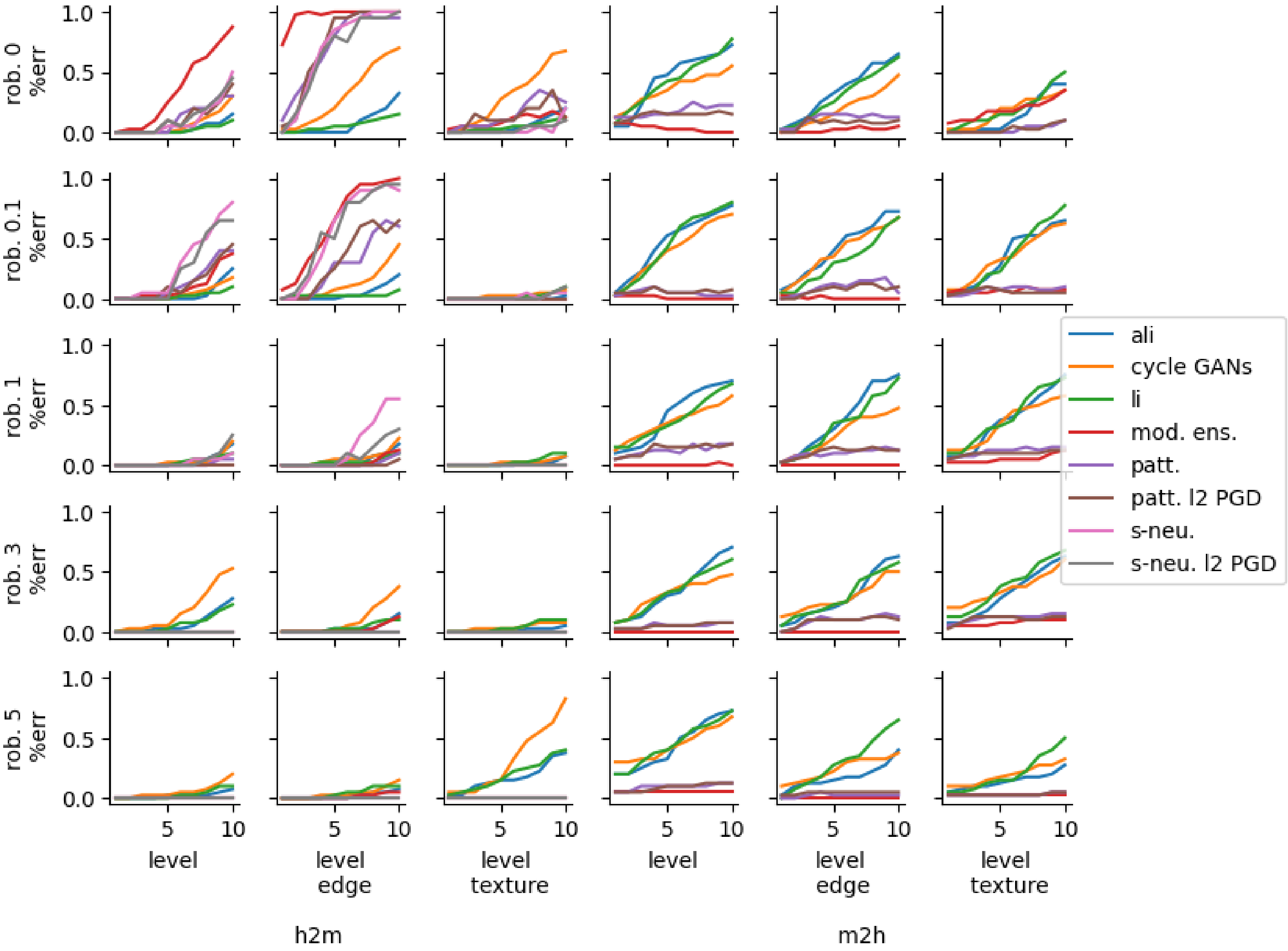


Figure 4.8: Misclassifications of the images produced bycite in function of the different noise levels. Each row represents the robustness of the network used to evaluate the adversarial images. The columns grouped by 3 represent the attack direction, h2m in the first 3, and m2h in the last 3, and inside these groups, there are 3 different fine-tuning, regular, edge data augmentation, and texture agnosticism. Each color line in the plots represents attack images created by[54] one particular model, which can be seen in the legend.

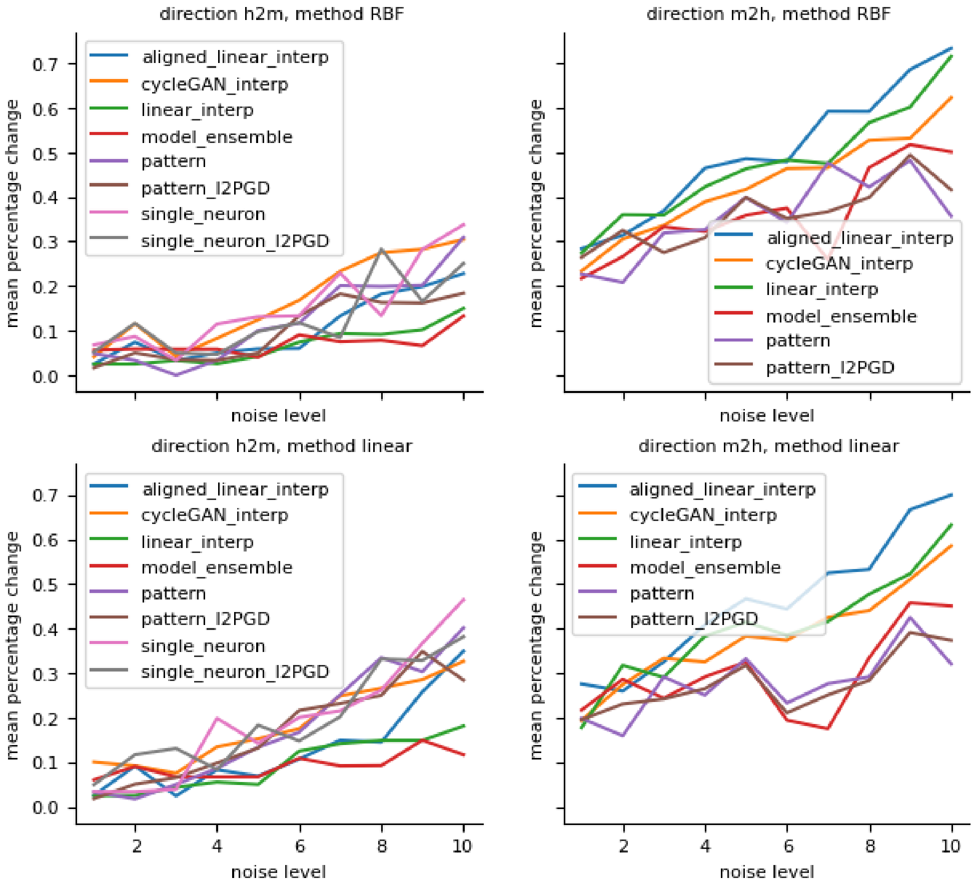


Figure 4.9: Mean misclassification rate of the macaque neural response in function of the pixel budget and attack methods: The first row is the percentage change using an RBF SVM, and the second one using a linear SVM. The columns represent the attack directions, h2m, and m2h. We can note the strong difference in misclassification between m2h and h2m starting from level 0.

4.6 Comparison to Neural and Behavioral Data:

4.6.1 Analyzing the Neural and behavioral data from Yuan et al.

To compare our models to the primate data, I analyzed the neural and behavioral data from [54] and present them here in a way compatible with the model results.

We observe in Fig 4.9 that the neural data reveal the same results regardless of whether a linear or RBF SVM was used, showing similar overall trend, and order of efficiency of the different methods.

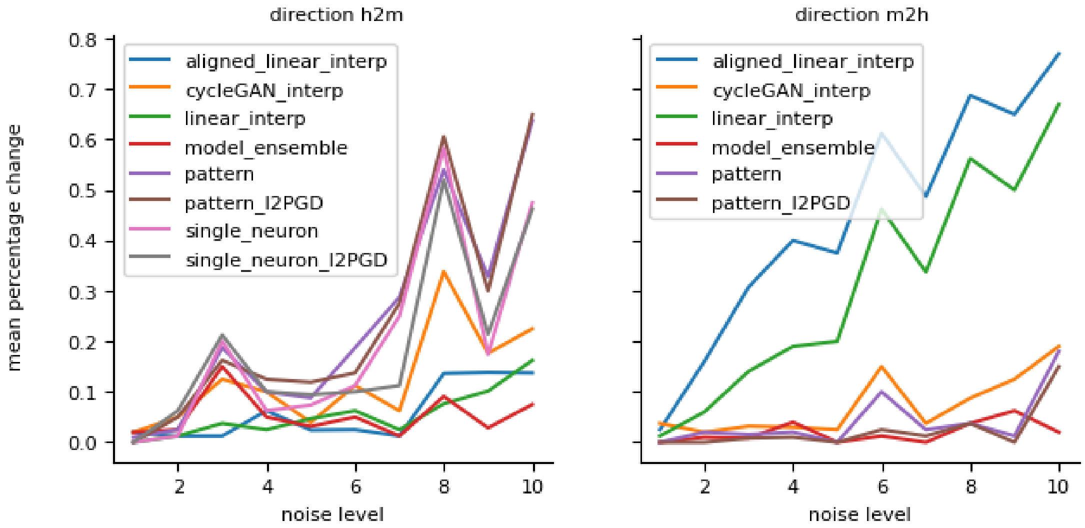


Figure 4.10: Mean misclassification rate of the human behavior trials in function of the pixel budget and attack methods: The first row is the percentage change using an RBF SVM, and the second one uses a linear SVM. The columns represent the attack directions, h2m, and m2h. Interestingly we do not have this gap between h2m and m2h at level 0.

In both cases, the h2m direction with low budget has a low success rate, inferior for all methods to 10%. The success rate grows slowly with the noise budget to reach 40% in the most successful method in the linear kernel, and 30% in the RBF SVM. We see that the linear interpolation and model ensemble are the least successful method in both SVMs barely reaching 10%. In the m2h direction, we also have a similar shape and curve organization across SVMs, with aligned linear interpolation, cycle GANs, and linear interpolations being the most successful method reaching 70% to 80% of percentage change in both cases at level ten. Overall it is clear that the m2h direction has a higher success rate starting from the lowest noise budget, and also has a steeper slope.

In plot 4.10 measuring the effect of the images on human behavior, we see that in both directions, images at a low noise budget are completely inefficient at fooling the human brain. In the h2m direction, we observe that the pattern and single neuron manage to reach the percentage of change up to 50%. This is interesting as it shows that a gray box attack based on a model of the macaque neural responses can efficiently fool the human brain. On the other hand, the linear interpolation, aligned linear interpolation, and model ensembles do not manage to reach more than 15% of the behavior change. For the m2h direction, we observe that only aligned linear interpolation and linear interpolation manage to fool the human brain to a large extent (over 70 %) whereas the graybox methods are not successful and only reach 10%(we only identified a neuron strongly responsive to monkey faces).

The cycle GAN method is in both cases slightly outperformed the worst methods but does not rival the best-performing ones and reaches at noise level 10, around 25%.

4.7 Correlating the models and primate data

We here compare the image creation techniques with a new objective. Until here we were evaluating how successful a method of creating images was to fool or not a model or primate, but we now compare the detailed error patterns of models.

We see in Fig 4.11 the correlation on image averaged levels between the human behavioral data and the monkey neural data across all pixel budgets and through two directions. In the h2m direction, we observe that the points are all aligned on the identity line, and despite some methods having variations around it, they are all well correlated. In the m2h direction, we observe that the points with the lowest efficiency are skewed and the monkey data is fooled much more than the human data, with no attacks being on average less than 20% efficient. We nonetheless see a strong correlation between human and monkey data and have a Pearson's r coefficient of 0.77.

We observe in 4.12 that in the m2h direction, the robustness and the training set of the

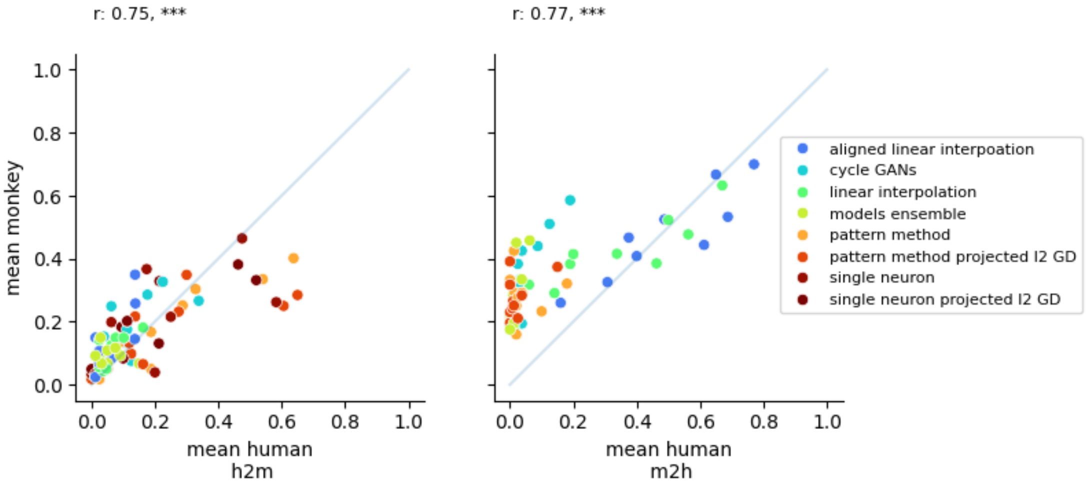


Figure 4.11: Correlations between human and monkey data misclassification data This figure shows the correlation across all pixel budgets of human and monkey results

network have very little effect on the performance, and the correlations between the models and the monkey neural data are very high. The strong correlation regardless of training or robustness is particularly impressive in the aligned linear interpolation, cycle GAN interpolation, and linear interpolation attacks.

The pattern and single neuron methods were less effective in fooling the models compared to fooling the monkey neural data. In all of the cases, the correlations for m2h attacks are above 0.74, and even reach up to 0.86 in the best-correlated cases. In the h2m direction, the attack images are not as successful in the monkey neural data, and the success rate varies a lot depending on the robustness and training set of the network. We observe that the attack images do not reach more than 50% disruption in the monkeys, making all points closer to the origin and more difficult to interpret.

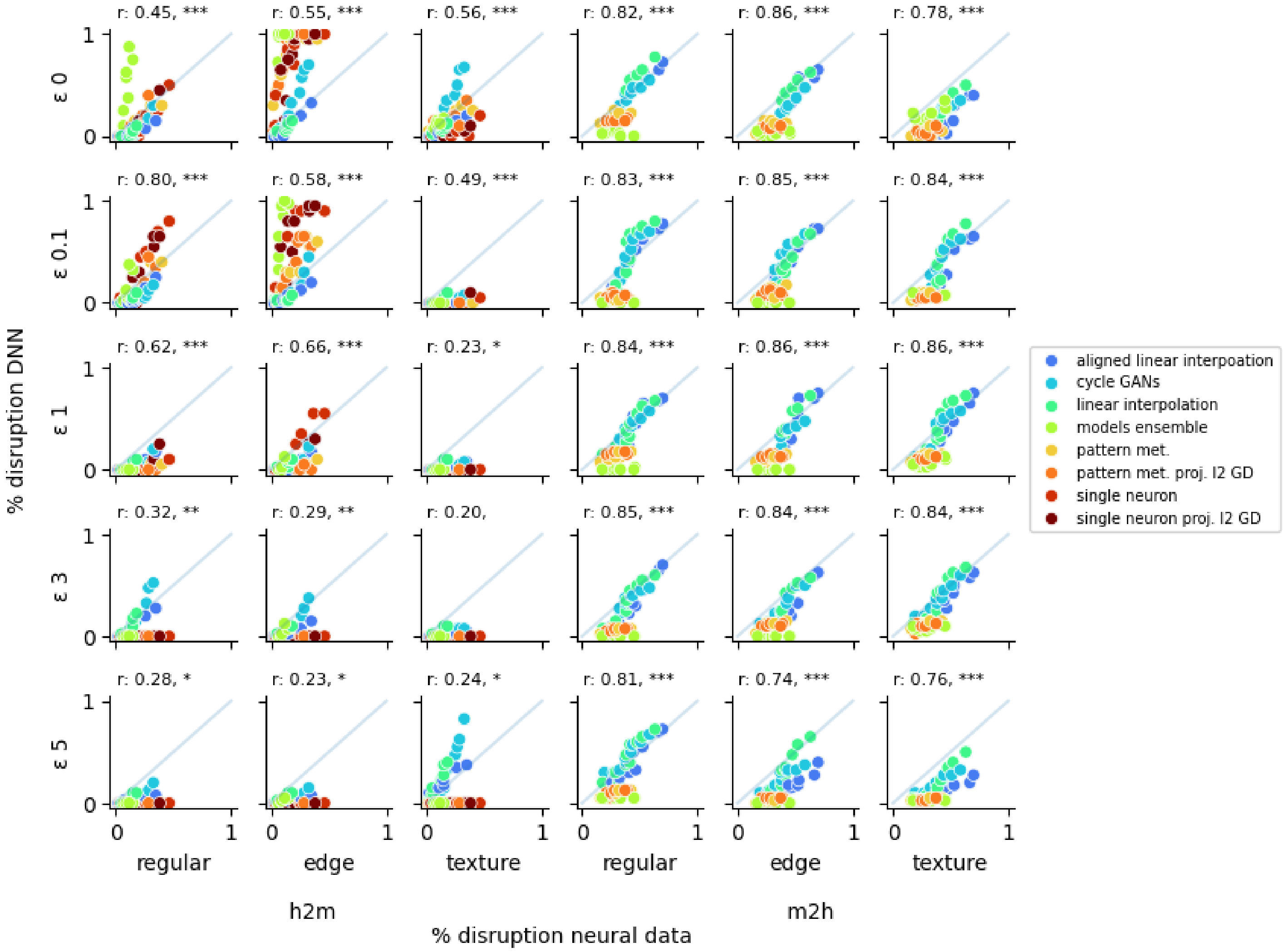


Figure 4.12: Percentage of misclassification of the neural network in function of the percentage of misclassification by the monkey neural data across all pixel budgets showing the correlations and it's statistical significance. Each row represents the robustness of the network used to evaluate the adversarial images. The columns grouped by 3 represent the attack direction, h2m in the first 3, and m2h in the last 3, and inside these groups, there are 3 different fine-tuning, regular, edge data augmentation, and texture agnosticism. Each color dot in the plots represents attack images created by one particular model, which can be seen in the legend. A surprising result is the similarity in the m2h correlations across all models.

For the regular models, we observe that the model ensemble is very efficient in fooling the vanilla model (robustness epsilon 0), and all the other methods are fooled as much as the non-robust model. In the 0.1 robustness case, the model is very correlated to the neural monkey data and reach the highest correlation, of 0.8. In all stronger robustness cases, the models gets less fooled by the images and the correlation therefore goes down with the most robust model reaching 0.28.

The edge models get fooled more than the regular model with attacks from all methods except for aligned linear interpolation that is over the identity line in both the epsilon 0 and epsilon 0.1 cases. The correlation for these models reaches a maximum in the epsilon 1 case where the efficiency of the attack methods is more aligned to the monkey neural data and reaches 0.66. After that, the models are more robust than the neural data and we can see that all the points are under the identity line.

The texture augmentation models are in the case of h2m the more robust to attacks, as the vanilla neural network performs similarly to the monkey neural data, and the adversarially trained neural networks are too robust and therefore not correlated. One outlier to this is the texture-agnostic robustness at level epsilon = 5 where the aligned linear interpolation and cycle GANs are effective at fooling the model in the same proportion as the monkey neural data.

In Fig 4.13 having a similar organization row and column-wise as the two previous ones, we can observe the correlation between model and neural data disruption rates across the different methods at only level 10. This allows us to see in more detail how each attack method is correlated by having 10 times fewer points. We chose level 10 as it is the most efficient one in fooling the models.

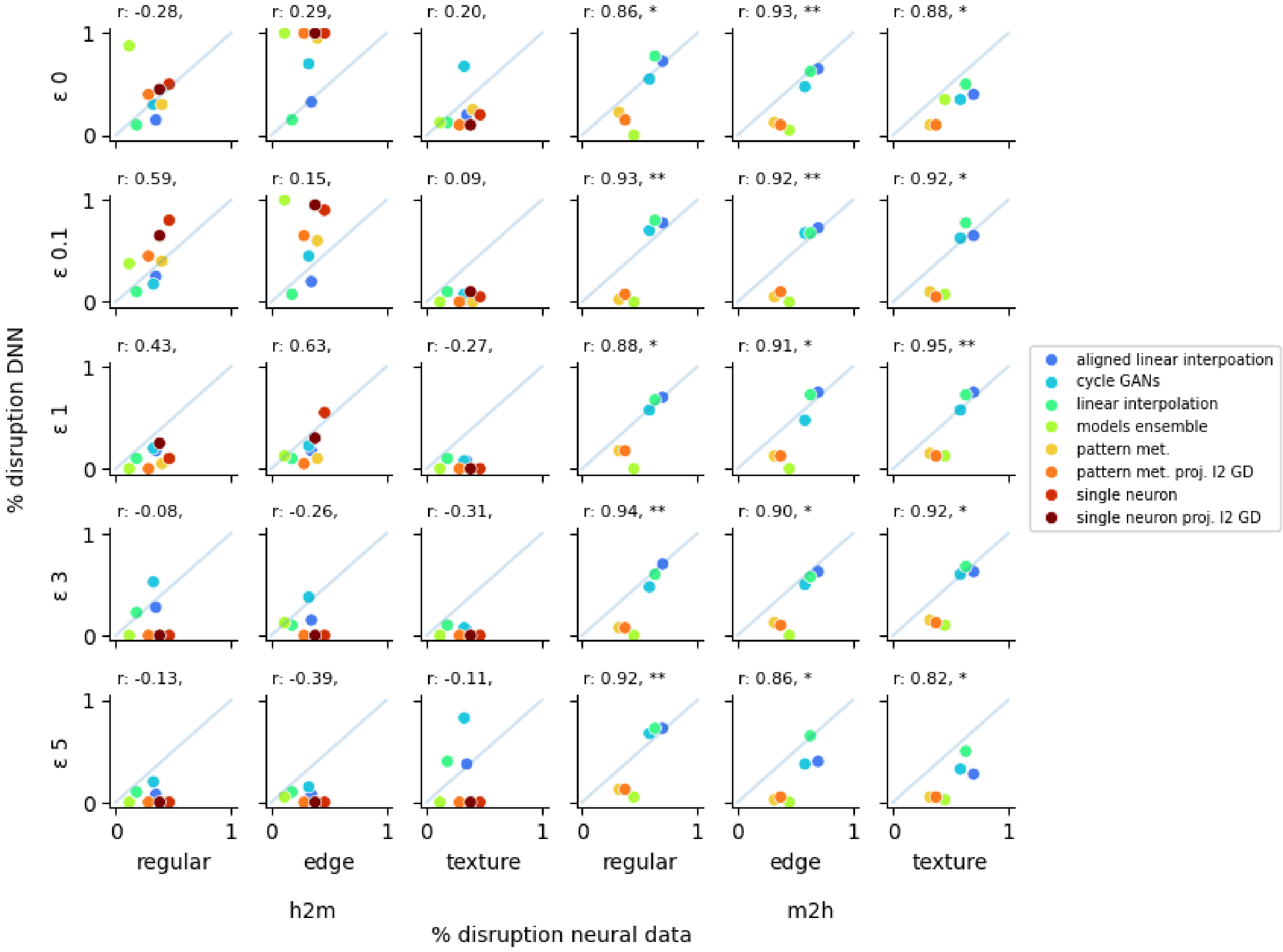


Figure 4.13: Percentage of misclassification of the neural network in function of the percentage of misclassification by the monkey neural data at pixel budget 10. This allows us to measure the Pearson correlation between our models and the neural responses shown in gray for each plot. Each row represents the robustness of the network used to evaluate the adversarial images. The columns grouped by 3 represent the attack direction, h2m in the first 3, and m2h in the last 3, and inside these groups, there are 3 different fine-tuning, regular, edge data augmentation, and texture agnosticism. Each color dot in the plots represents attack images created by one particular model, which can be seen in the legend. Level 10 allows to have a clearer view of one attack method's correlations using the value that is the most efficient.

In the h2m direction, we observe overall much lower correlations than in the m2h direction, however, there is a high variance in the causes of the low correlation. Indeed, we can see that the regular and edge models at robustness epsilon = 0 and 0.1 get fooled more easily than the macaque neural data (points are above the identity line), but the stronger adversarial networks are much more robust to the attacks and have the points under the identity line, both leading to low correlation. The texture agnostic model in the h2m direction is overall more robust in all cases than monkey vision and performs poorly on all robustness levels.

In the m2h direction, there is a much stronger correlation with the lowest being 0.82 and the highest 0.95. We mostly observe that the 3 attack methods where the model was susceptible (linear aligned linear and GAN interpolations) are very well correlated with the neural data. In the other methods the models are more robust to the attacks than the macaque vision, however, since the macaques are not very susceptible to the attacks it does not impact a lot the correlation.

In Fig 4.14 showing the correlation between the models and behavioral human data, we see in the m2h direction that the pattern method and the model ensemble are very inefficient for both human behavior and model. On the other hand, aligned linear interpolation, cycle GAN, and linear interpolation are very effective and correlated in almost all cases, bringing overall high correlations. In the h2m direction, we observe very similar results to the monkey neural data case with a higher variance of the points around the identity line. The regular vanilla model is fooled a lot by the model ensemble that has very little effect on human behavior, and the other attack methods are as effective in the models as in humans. The most correlated regular model is at robustness epsilon = 0.1 with correlation 0.68. In the more robust cases, the correlation even goes down until reaching 0.01.

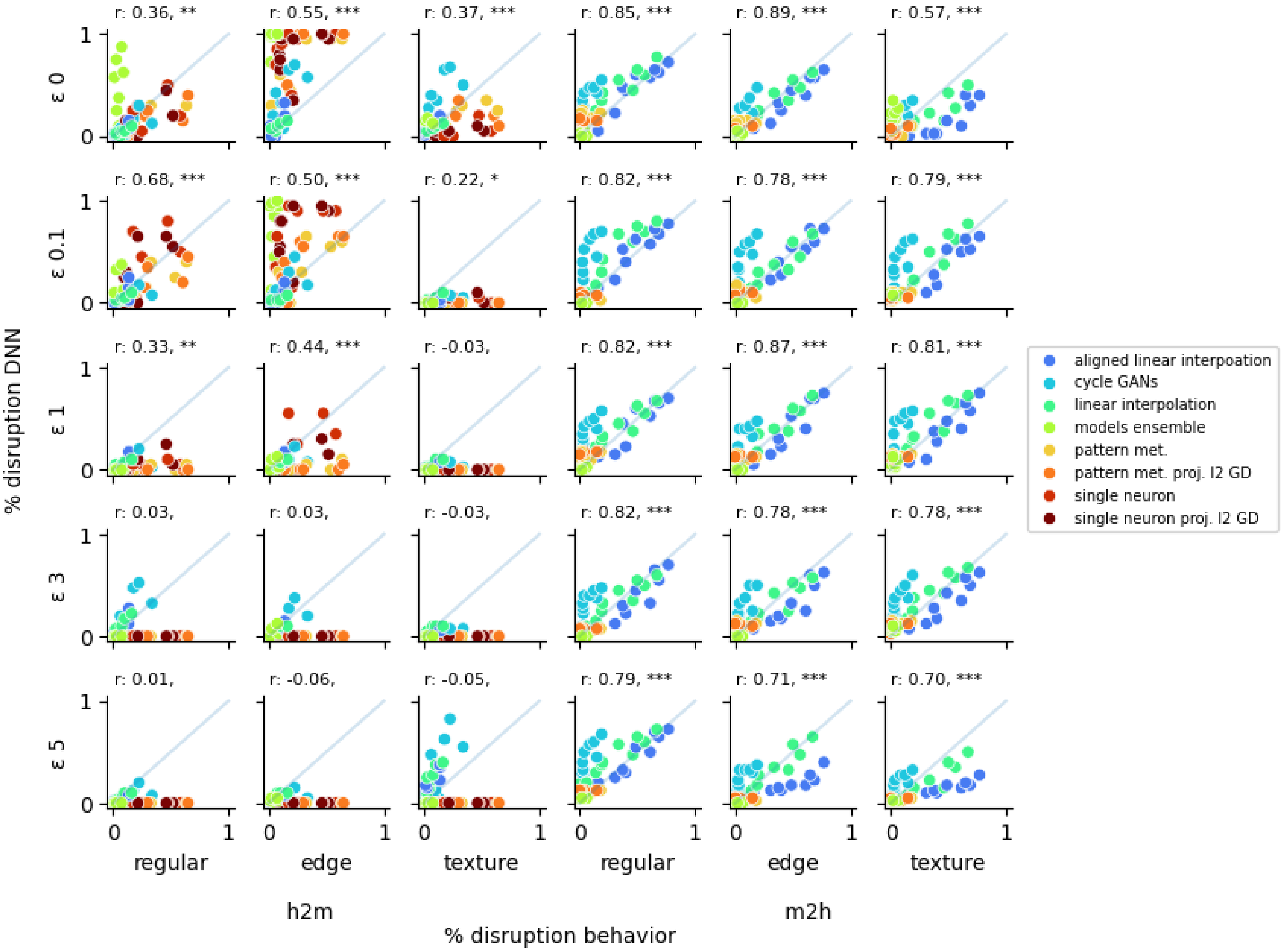


Figure 4.14: Percentage of misclassification of the neural network in function of the percentage of misclassification by the human behavioral data at all noise levels. This allows us to measure the Pearson correlation between our models and the neural responses shown in gray for each plot. Each row represents the robustness of the network used to evaluate the adversarial images. The columns grouped by 3 represent the attack direction, h2m in the first 3, and m2h in the last 3, and inside these groups, there are 3 different fine-tuning, regular, edge data augmentation, and texture agnosticism. Each color line in the plots represents attack images created by one particular model, which can be seen in the legend.

Similarly, in the edge training set, the model gets fooled a lot at low robustness compared to human vision, with all the points in the first two epsilon being above the identity line. As the robustness goes up the points pass under the identity line and the model does not get fooled anymore. We even reach a negative correlation at epsilon 5.

In the texture case, we have similar results to the neural data. The vanilla model is the most correlated with human behavior, but with the growing robustness, we have a decreasing correlation that reaches -0.05 at epsilon 5. We also have outlier data from the aligned linear interpolations and linear interpolations being effective against the epsilon = 5 texture model even though it is not against the less robust models.

In this figure 4.15 showing correlations of models and behavior using only the level 10 data, we can more clearly see which attack method is efficient and which isn't. In the m2h direction, we can see how the aligned and regular linear interpolation are almost as effective in all cases, and how the pattern and model ensemble are not efficient in fooling the models and the human behavior. We can also see that the cycle GANs are overall more efficient in the models but still manage to fool humans to some extent.

The h2m direction shows similar results to the monkey-model correlation, with the attack methods being efficient in the low robustness cases but not fooling the models with higher robustness.

Figure 4.16 summarizes the correlations depending on the training set used across both behavior and neural data, at all pixel budgets and at level 10. We can see that the m2h direction is in all cases much more successful than the h2m direction. These plots allow us to select what models seem to be the most promising to model the visual responses to adversarial attacks in primates.

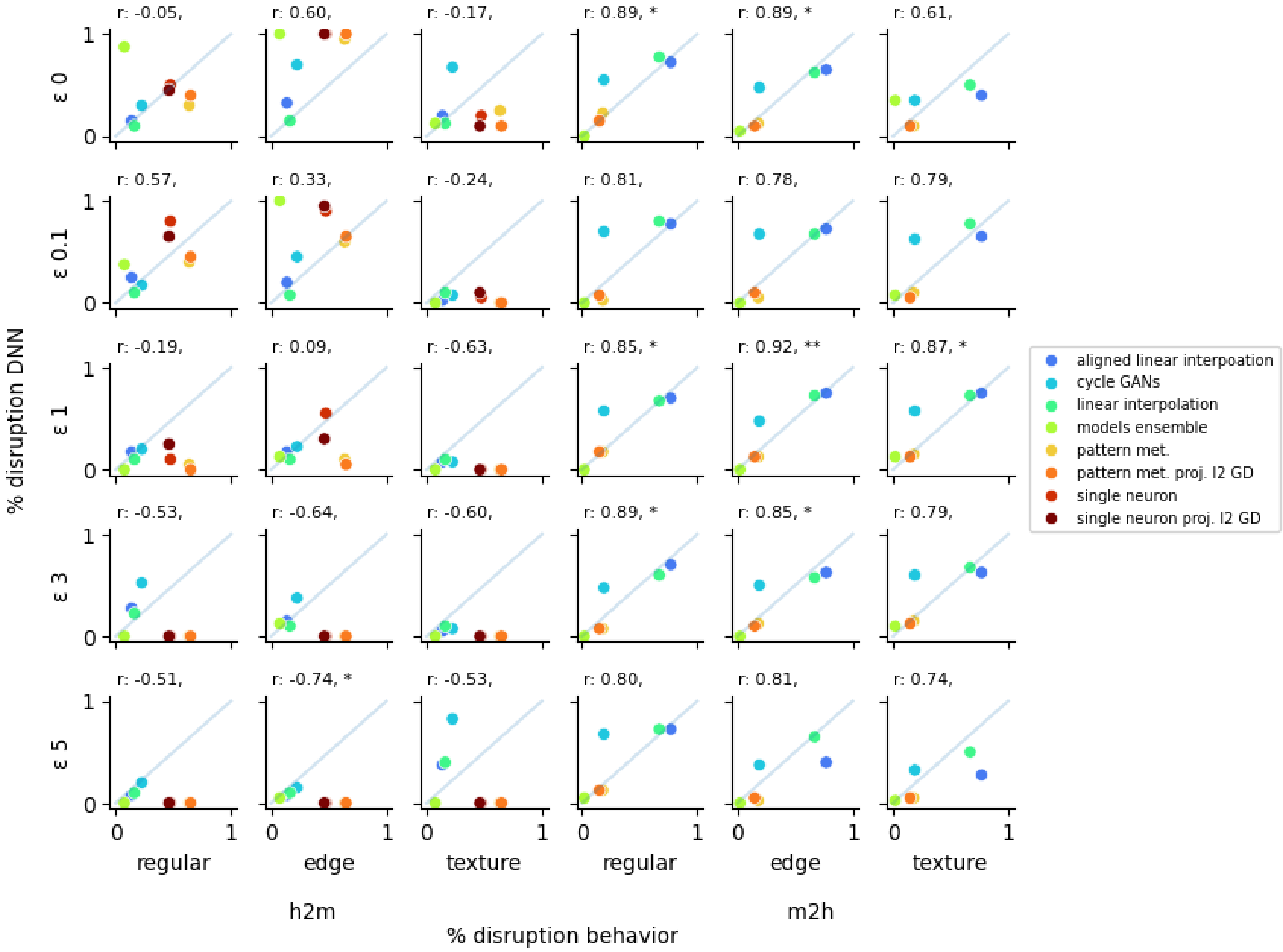


Figure 4.15: Percentage of misclassification of the neural network in function of the percentage of misclassification by the human behavioral data at pixel budget 10. This allows us to measure the Pearson correlation between our models and the neural responses shown in gray for each plot. Each row represents the robustness of the network used to evaluate the adversarial images. The columns grouped by 3 represent the attack direction, h2m in the first 3, and m2h in the last 3, and inside these groups, there are 3 different fine-tuning, regular, edge data augmentation, and texture agnosticism. Each color line in the plots represents attack images created by one particular model, which can be seen in the legend.

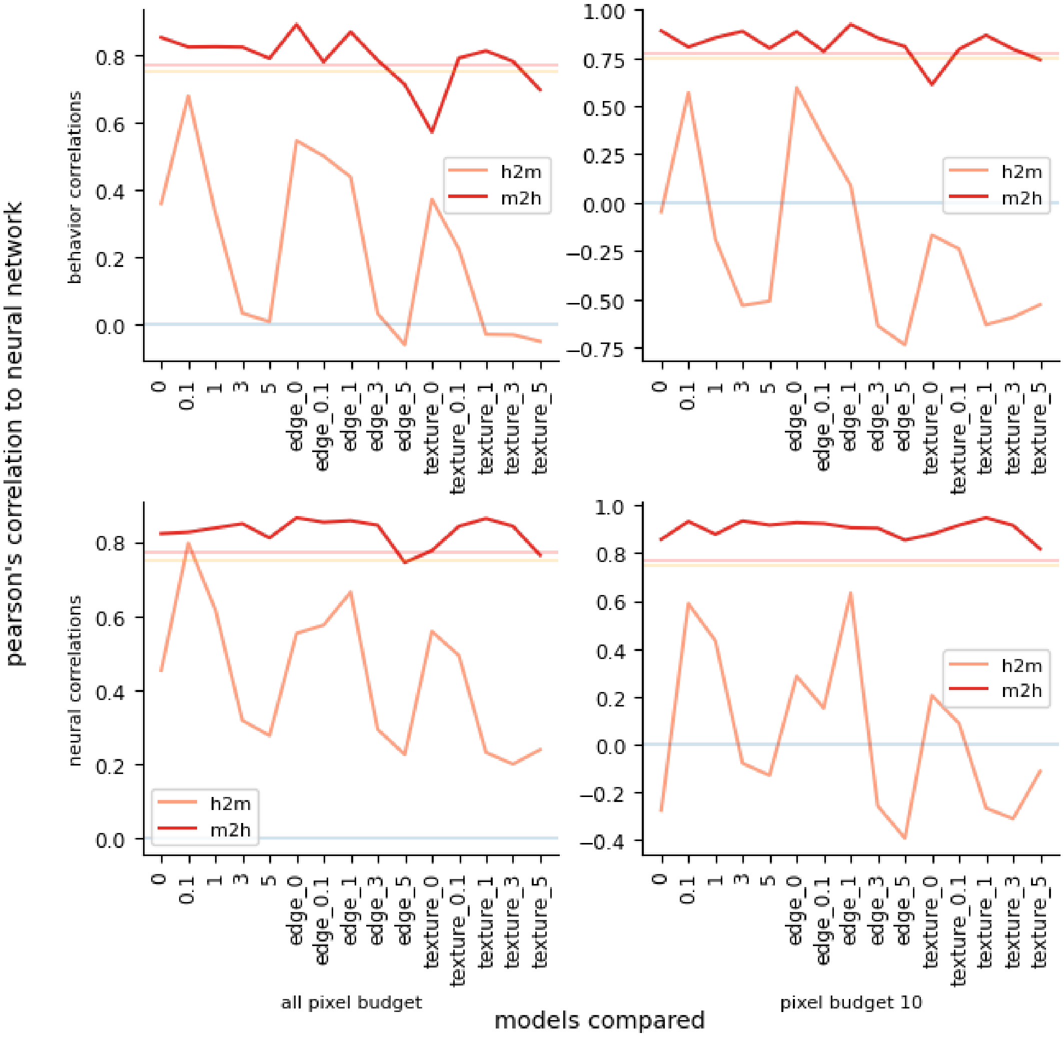


Figure 4.16: Pearson's correlations depending on the model and direction The first row is the human behavior data, and the second one is the neural data. The first column is the correlation across all pixel budgets and the second one is only pixel budget 10. The person correlations shown in the horizontal line represent how correlated human and macaque data are. The correlations therefore allow us to see how closely a model is correlated to the primate vision compared to how to primates are related.

We can therefore say that across both data type, the regular model with robustness 0.1 is the best at modeling, followed by the edges 0.1 and 1 based on their correlation with primate data. We can also say that strong adversarial training is not a good solution as it brings a lower correlation, especially in the h2m direction.

Using this figure 4.16 we can also easily compare our models to human and monkey correlations. We have Pearson's R for h2m of 0.75 and m2h 0.77 in the neural-behavioral correlation case. Therefore we can see that model-human behavior correlation using the 0.1 robustness regular model comes very close to the human-monkey correlation in the h2m direction(r = 0.68), and surpasses it in the m2h case (r = 0.82) and reaches even higher in the m2h edge epsilon 1 with r = 0.87. In the model-neural behavior case, the correlations reach even higher with r= 0.8 in the h2m direction for the regular model and epsilon 0.1 and r>= 0.74 in the m2h direction with the regular model and epsilon 0.1, r = 0.83. Therefore, we can affirm that at least the regular model with robustness = 0.1 is more closely correlated to monkey neural data than the monkey-to-human behavior for adversarial classification tasks. The model-to-human achieves similar correlation strength compared to the human-to-monkey.

4.8 Investigation of the attack efficiency imbalance between the direction (followed)

We plotted the hyperplane distance instead of the success rate to the data from[54] to see the differences across different levels, as we did not have this data in the pilot studies.

We can see in Fig 4.17 which shows the distance to the SVM hyperplane that the imbalance in the success rate is present.

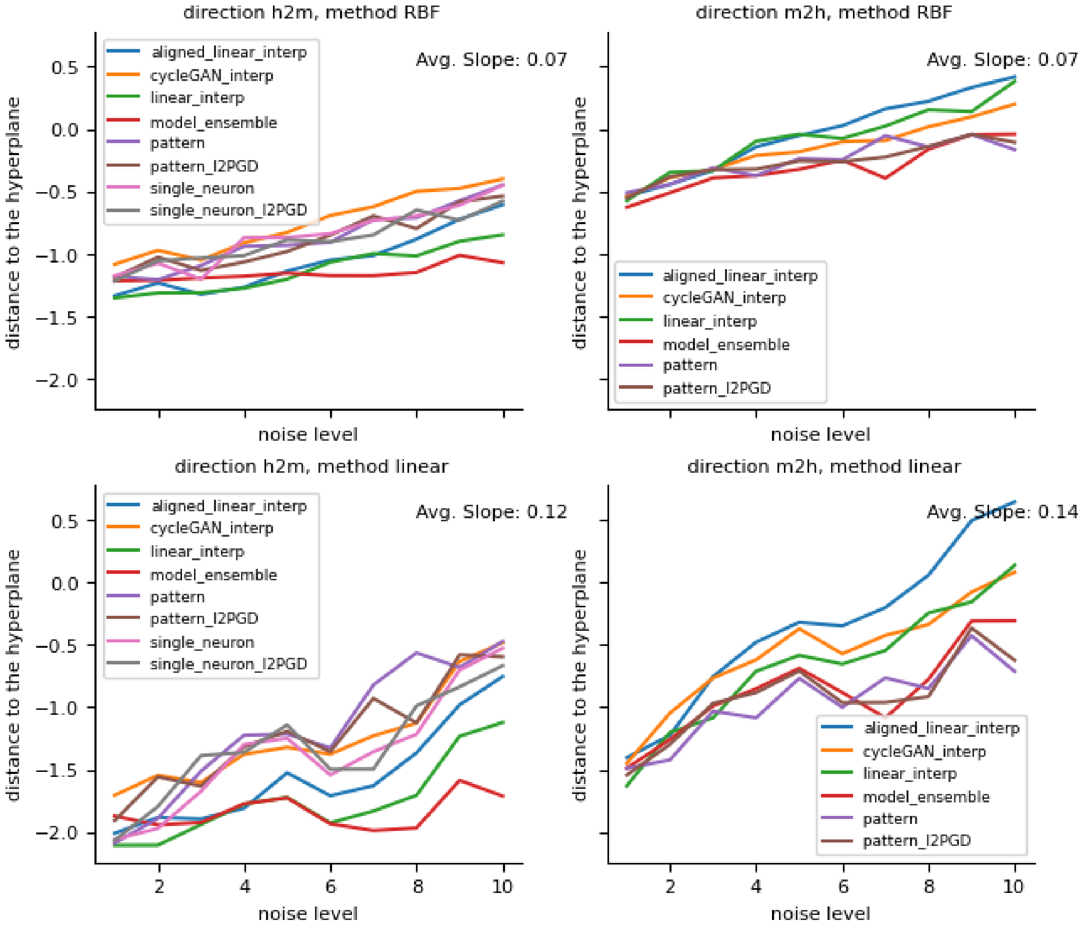


Figure 4.17: Mean distance of the test data to the SVM hyperplane in function of the noise level and the attack method The first row represents the RBF SVM and the second row represents the linear SVM. The first column represents the h2m direction and the second row the m2h. a negative distance represents that a point is still on its original side of the hyperplane, whereas a positive distance is a point that is being classified on the other side of the SVM hyperplane. We can see the average slope across all attacks, and we observe that the only difference in crossing the hyperplane is the origin and not the linear dependence on the pixel budget.

However, we can see that the average slope of the distance to the SVM hyperplane in function of the level is almost identical, and only the original distance of the points to the hyperplane changes. This means that the attack methods are in fact as efficient in both directions, but that somehow the m2h direction is closer to the hyperplane at the same noise level. Therefore, if we interpolate this curve to noise level 0, the monkey images should be without any attack closer to the SVM hyperplane.

To investigate this we would need to plot the distance of the clean test images to the SVM hyperplane, however, as macaque experiments are expensive, we do not have the neural results of the original test images.

Because of this, we investigated if our hypothesis holds in the model representations in fig 4.18. We passed through the models the training images that are used to create the SVM hyperplane and the test images that are then measured against the SVM hyperplane. We took the activations from the last convolutional layer being 224*224, normalized the output and then performed a PCA decomposition to get the first two principal components. We then projected the training and test data to these two PCs and plotted the results.

We can see that the human face train and test dataset are completely overlapping, however, we see that the training set and the test set of monkey faces are not in the same distribution. Indeed the test set of monkey faces overlaps the PC projection of the human faces a lot more than the training faces, and the test set also does not overlap a large space spanned by the training set.

This confirms the hypothesis that the test set of monkey faces is closer to the human training set and therefore closer to the SVM hyperplane in the neural network space. Therefore, with the fact that both directions have the same slopes in function of the pixel budget of the

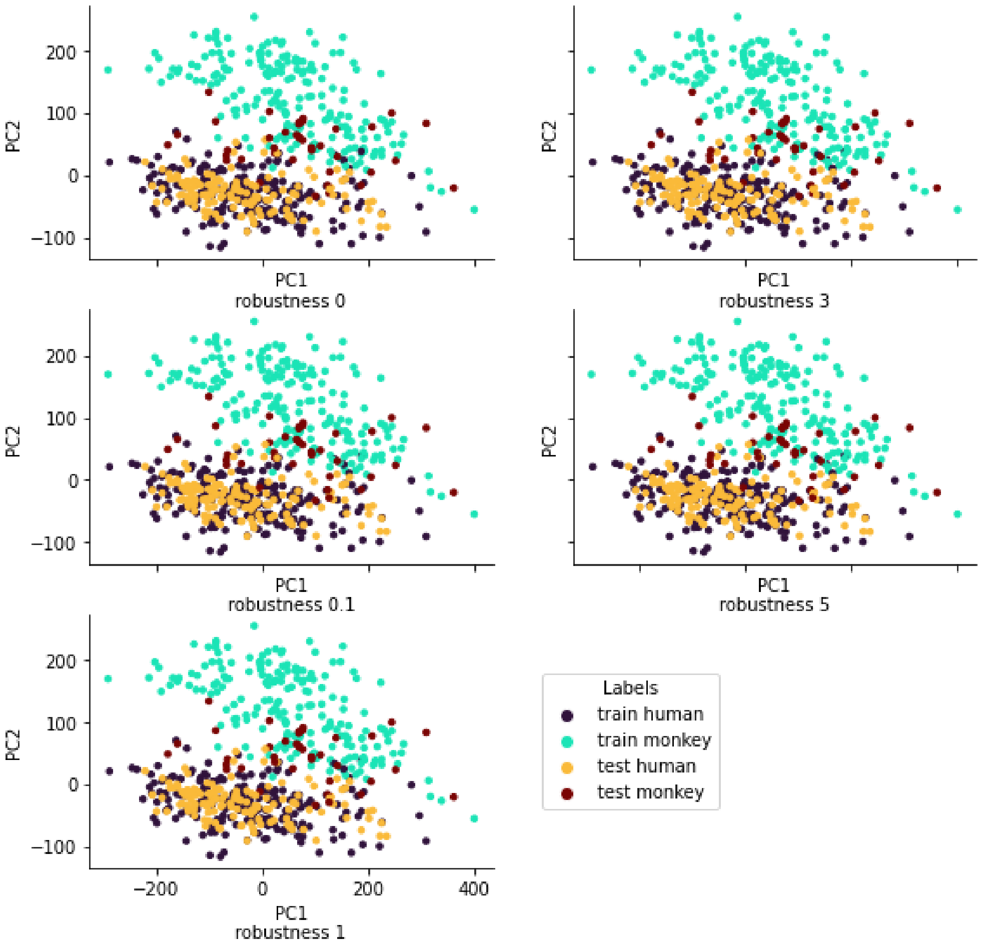


Figure 4.18: Projection of the regular model's last convolutional layer activations into the two main principal components in function of different datasets. The five plots represent the 5 different robustness levels of the regular ResNet50. The points are the activation of the last convolutional layer activation projected in two dimensions using the PCA decomposition of the training data. The goal of the PCA is to have a visualization of the largest variance in the data, this allows us to observe if the training and testing sets are overlapping as they should or if they are seemingly from two distributions.

attacks, there is a strong argument for believing that the distribution imbalance also holds in the monkey neural space. If this is true, it is logical that we observe a stronger success rate in the adversarial attack.

5

Discussion and future directions

5.1 Discussion

Our findings can be summarized as follows:

We successfully replicated the training results from[15] using a face dataset instead of ImageNet. This not only validates our methods but also extends the findings of[15] to controlled environments with binary choices. Interestingly, our model exhibited greater robustness than expected from its adversarial training. This could be attributed to the binary choice setup, as opposed to the thousand-category classification in the original study. Alternatively, it may result from fine-tuning an ImageNet-robust model on our new dataset as Imagenet has a macaque category but no human one. Interestingly we also found out that the adversarial training mostly impacts the h2m direction in the models we have used.

Our results align with Guo et al.[19], demonstrating that robust models can match or exceed human vision in terms of robustness, depending on the chosen robustness level and data augmentation techniques. This result is moreover on a more controlled task using a 2-way classification instead of a 1000-way classification, confirming the result shown that it is possible to build a model as robust as primate vision. However, this finding requires careful interpretation, as shown by[54], macaques can be trained to become more robust. Moreover, methods such as pattern, single neuron, and model ensemble are designed to fool neural networks in ways similar to how robust neural networks are trained, potentially explaining their resistance to certain attacks. On the other hand, our pilot studies making adversarial images from the robust neural networks did not yield great efficiency at fooling the macaques, especially in the h2m direction.

Correlating our results with human and monkey data, we found an optimal robustness of 0.1 using the regular training model. This suggests that primates may naturally develop adversarial robustness equivalent to an epsilon of 0.1, possibly due to exposure to challenging visual conditions (e.g., heavy rain, fog) or other factors. Alternatively, this robustness could stem from visual capabilities not explicitly trained in our models, such as partial object occlusion or 3D spatial perception. Further investigation into these possibilities may require new adversarial training approaches.

Comparing the model correlations to the correlation between human behavior and monkey neural data, we saw that the 0.1 model was more correlated to the monkey neural data than primate data were together. It also reaches close performances to the human data, human-model correlation outperforming the inter-primate correlation in one direction but not in the other. This result shows that the adversarially trained neural network can rival primate vision at least on this set of attacks.

Contrary to our initial hypothesis, we found that the imbalance between attack directions is not caused by the model's focus on texture alone. We had theorized that blurring facial hair would suffice to shift classification towards the human category, while adding hair texture to human faces would be too costly in terms of pixel budget. However, our second pilot experiment disproved this theory, showing no significant differences between the regular model, edge data augmentation, and texture-agnostic model.

We identified the cause of the directional efficiency difference by analyzing the distance to the hyperplane. Monkey face attack images were closer to the SVM margin than human face images, although both directions increased linearly at the same rate with increased pixel budget. This suggested either a linearity caused by luck at the tested noise levels or a discrepancy in the original test images. PCA analysis of the training set used for SVM creation, with subsequent projection of the test set, revealed that the test image neural network representations are not in the same distribution as the training images, explaining the observed imbalance. Another hypothesis is that the model, trained on imagenet with a macaque category, has a wider internal representation of macaques explaining the fig 4.18, and the cause for the similar slope is still to be determined.

To address this issue, one potential solution is to use noise level 1 for creating the SVM. As shown in Fig. 4.18, the test sets remain reasonably separable at this level. While this approach assumes that noise level 1 attacks do not fool the monkey, it offers a way forward without additional data collection. Supporting this approach, Fig. 4.10 shows 0% monkey-to-human misclassification at noise level 0, compared to 20% in Fig. 4.9, suggesting that neural data might also exhibit 0% misclassification at this level.

5.2 Next steps

The next crucial step is to achieve complete reproducibility of the images through attacks using the robustness library. This is a critical bottleneck, as it currently prevents us from proceeding to more resource-intensive experiments, namely collecting neural data from the macaques and behavioral data via Mechanical Turk.

In parallel with developing attack images, I've begun setting up an Amazon Mechanical Turk experiment. While it functions in the sandbox environment, it crashes upon release to production. Resolving this issue is essential for collecting human behavioral data. Additionally, I need to conduct toy experiments to establish a pipeline for retrieving and analyzing data from MTurk.

Regarding macaque neural data, we've already completed two pilot experiments. By combining this with data from previous images, I've developed a functional pipeline ready for data analysis.

Once these preliminary steps are complete, we can proceed with data collection and analysis.

Looking further ahead, I'm interested in exploring the creation of adversarial images using a diffusion model, as an alternative to the iterative fast gradient sign method that follows the model's gradient. I believe this approach could generate a new class of more realistic adversarial images. This idea seems realizable as diffusion models have already been used to create attack images in the context of fooling a face recognition CNN while changing minimally the face[31]. This implementation would obviously need some engineering to be transformed for a new goal, however, it shows that it is possible to use a diffusion model to create adversarial images. If successful, it could potentially be applied directly to create adversarial images from neural data, eliminating the need for a neural network as a brain model for attacks. This approach offers two key advantages: 1) attacks that are less prone to overfitting artificial neural network weaknesses, and 2) the ability to train DNNs on biological adversarial data, potentially yielding more accurate models of primate vision.

I feel truly fortunate to be able to continue this project with Will in the Livingstone lab and take on some new ones during the course of the next year to continue combining neuroscience and machine learning. I once again want to thank everyone who has had a part in making this thesis possible, it has truly been a great experience.

References

[1] Afraz, S.-R., Kiani, R., & Esteky, H. (2006). Microstimulation of inferotemporal cortex influences face categorization. *Nature*, 442(7103), 692–695.

[2] Alain, G., Bengio, Y., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2017). UNDERSTANDING INTERMEDIATE LAYERS USING LINEAR CLASSIFIER PROBES. *International Journal of Computer Vision*, 115(3), 211–252.

[3] Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., & DiCarlo, J. J. (2014). Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. *PLoS Computational Biology*, 10(12), e1003963.

[4] Carlini, N. & Wagner, D. (2017). Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. arXiv:1705.07263 [cs].

[5] Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., & Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 6(1), 27755.

[6] Cox, D. D. & Savoy, R. L. (2003). Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19(2), 261–270.

[7] Dapello, J., Marques, T., Schrimpf, M., Geiger, F., Cox, D. D., & DiCarlo, J. J. (2020). Simulating a Primary Visual Cortex at the Front of CNNs Improves Robustness to Image Perturbations.

[8] Deng, J. & Dong, W. (2009). ImageNet: A Large-Scale Hierarchical Image Database.

[9] DiCarlo, J., Zoccolan, D., & Rust, N. (2012). How Does the Brain Solve Visual Object Recognition? *Neuron*, 73(3), 415–434.

[10] 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. arXiv:2010.11929 [cs].

[11] Elsayed, G. F., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., & Sohl-Dickstein, J. (2018). Adversarial Examples that Fool both Computer Vision and Time-Limited Humans. arXiv:1802.08195 [cs, q-bio, stat].

[12] Ettinger, A., Elgohary, A., & Resnik, P. (2016). Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP* (pp. 134–139). Berlin, Germany: Association for Computational Linguistics.

[13] Francioni, V., Tang, V. D., Brown, N. J., Toloza, E. H., & Harnett, M. (2023). Vectorized instructive signals in cortical dendrites during a brain-computer interface task.

[14] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A Neural Algorithm of Artistic Style. arXiv:1508.06576 [cs, q-bio].

[15] Gaziv, G., Lee, M. J., & DiCarlo, J. J. (2023). Robustified ANNs Reveal Wormholes Between Human Category Percepts. arXiv:2308.06887 [cs, q-bio].

[16] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2022). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv:1811.12231 [cs, q-bio, stat].

[17] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. arXiv:1406.2661 [cs, stat].

[18] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [cs, stat].

[19] Guo, C., Lee, M. J., Leclerc, G., Dapello, J., Rao, Y., Madry, A., & DiCarlo, J. J. (2022). Adversarially trained neural representations may already be as robust as corresponding biological neural representations.

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). Las Vegas, NV, USA: IEEE.

[21] Hebb, D. (1949). The_organization_of_behavior.pdf.

[22] Huang, X. & Belongie, S. (2017). Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. arXiv:1703.06868 [cs].

[23] Jang, H., McCormack, D., & Tong, F. (2021). Noise-trained deep neural networks effectively predict human vision and its neural responses to challenging images. *PLOS Biology*, 19(12), e3001418.

[24] Kamitani, Y. & Tong, F. (2005). Decoding the visual and subjective contents of the human brain. *Nature Neuroscience*, 8(5), 679–685.

[25] Khaligh-Razavi, S.-M. & Kriegeskorte, N. (2014). Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Computational Biology*, 10(11), e1003915.

[26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.

[27] Kubilius, J., Schrimpf, M., Kar, K., Rajalingham, R., Hong, H., Majaj, N., Issa, E., Bashivan, P., Prescott-Roy, J., Schmidt, K., Nayebi, A., Bear, D., Yamins, D. L., & DiCarlo, J. J. (2019). Brain-Like Object Recognition with High-Performing Shallow Recurrent ANNs.

[28] Kurakin, A., Goodfellow, I., & Bengio, S. (2017). Adversarial Machine Learning at Scale. arXiv:1611.01236 [cs, stat].

[29] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

[30] Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6), 335–346.

[31] Liu, J., Lau, C. P., & Chellappa, R. (2023). DiffProtect: Generate Adversarial Examples with Diffusion Models for Facial Privacy Protection. arXiv:2305.13625 [cs].

[32] Logothetis, N. K., Pauls, J., & Poggio, T. (1995). Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5), 552–563.

[33] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2019). Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [cs, stat].

[34] MCCULLOCH, W. (1943). A logical calculus of the ideas immanent in nervous activity.

[35] Mollon, J. D. & Bowmaker, J. K. (1992). The spatial arrangement of cones in the primate fovea. *Nature*, 360(6405), 677–679.

[36] Nowak, L. (1997). The timing of information transfer in the visual system.

[37] Orban, G. A., Van Essen, D., & Vanduffel, W. (2004). Comparative mapping of higher visual areas in monkeys and humans. *Trends in Cognitive Sciences*, 8(7), 315–324.

[38] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical Black-Box Attacks against Machine Learning. arXiv:1602.02697 [cs].

[39] Rajalingham, R. & DiCarlo, J. J. (2019). Reversible Inactivation of Different Millimeter-Scale Regions of Primate IT Results in Different Patterns of Core Object Recognition Deficits. *Neuron*, 102(2), 493–505.e5.

[40] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.

[41] Rumelhart, D. E., Hintont, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors.

[42] Sacramento, J., Costa, R. P., Bengio, Y., & Senn, W. (2018). Dendritic cortical microcircuits approximate the backpropagation algorithm. arXiv:1810.11393 [cs, q-bio].

[43] Schmolesky, M. T., Wang, Y., Hanes, D. P., Thompson, K. G., Leutgeb, S., Schall, J. D., & Leventhal, A. G. (1998). Signal Timing Across the Macaque Visual System. *Journal of Neurophysiology*, 79(6), 3272–3278.

[44] Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Geiger, F., Schmidt, K., Yamins, D. L. K., & DiCarlo, J. J. (2018). Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like?

[45] Simonyan, K. & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs].

[46] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567 [cs].

[47] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. arXiv:1312.6199 [cs].

[Tanaka] Tanaka, K. INFEROTEMPORAL CORTEX AND OBJECT VISION.

[49] Tootell, R., Silverman, M., Hamilton, S., Switkes, E., & De Valois, R. (1988). Functional anatomy of macaque striate cortex. V. Spatial frequency. *The Journal of Neuroscience*, 8(5), 1610–1624.

[50] Van Essen, D. C., Newsome, W. T., & Maunsell, J. H. (1984). The visual field representation in striate cortex of the macaque monkey: Asymmetries, anisotropies, and individual variability. *Vision Research*, 24(5), 429–448.

[51] Vanduffel, W., Fize, D., Peuskens, H., Denys, K., Sunaert, S., Todd, J. T., & Orban, G. A. (2002). Extracting 3D from Motion: Differences in Human and Monkey Intraparietal Cortex. *Science*, 298(5592), 413–415.

[52] Veerabadran, V., Goldman, J., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., Shlens, J., Sohl-Dickstein, J., Mozer, M. C., & Elsayed, G. F. (2023). Subtle adversarial image manipulations influence both human and machine perception. *Nature Communications*, 14(1), 4933.

[53] Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23), 8619–8624.

[54] Yuan, L., Xiao, W., Dellaferrera, G., Kreiman, G., Tay, F. E. H., Feng, J., & Livingstone, M. S. (2022). Fooling the primate brain with minimal, targeted image manipulation. arXiv:2011.05623 [cs, eess, q-bio].

[55] Zhou, Z. & Firestone, C. (2019). Humans can decipher adversarial images. *Nature Communications*, 10(1), 1334.