Invariant Object Recognition in Deep Neural Networks and Humans

Haider Al-Tahan, Western University

Supervisor: Mur, Marieke, The University of Western Ontario

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Neuroscience
© Haider Al-Tahan 2023

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Computational Neuroscience Commons

Recommended Citation

https://ir.lib.uwo.ca/etd/9894

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.
Abstract

Invariant object recognition, a cornerstone of human vision, enables recognizing objects despite variations in rotations, positions, and scales. To emulate human-like generalization across object transformations, computational models must perform well in this aspect. Deep neural networks (DNNs) are popular models for human ventral visual stream processing, though their alignment with human performance remains inconsistent. We examine object recognition across transformations in human adults and pretrained feedforward DNNs. DNNs are grouped by architecture, visual diet, and learning goal. We focus on object rotation in depth, and observe that object recognition performance is better preserved in humans than in DNNs, although they show a similar pattern in how performance drops as a function of rotational angle. DNNs also exhibit decreased recognition after other transformations, especially scale changes. Model architecture minimally influences performance, while DNNs trained on richer visual diets and semi-supervised learning goals excel. Our study suggests that visual diet and learning goals may play an important role in the development of invariant object recognition in humans.

**Keywords:** Deep Neural Networks, Invariant Object Recognition, Ventral Visual Stream
Humans excel at identifying objects they see, regardless of the object’s position, location, or size. We aim to bridge the gap between humans and machines, both to improve algorithms of visual object identification and to gain insights into how the human brain performs visual recognition. A specific kind of algorithms, known as Deep Neural Networks (DNNs), aims to replicate the way human vision operates. Various DNNs, already trained on image recognition, were put to the test in a task that mimics human visual challenges. These DNNs were categorized based on their design, their training goals, and the types of images they had been trained on. Results from comparing DNNs to humans revealed that the design of DNNs was less important than the diversity and quality of the images they were trained on. Furthermore, the way we train DNNs; whether supervised, unsupervised, or semi-supervised, plays a crucial role for how well DNNs can identify objects. Lastly, we found that even the top-performing DNNs fell short of human capabilities in identifying objects under different naturalistic conditions.
Dedication

The achievement of my master’s degree is a shared triumph, made possible by the incredible people who stood by me, offering motivation and assistance. It is to these remarkable individuals that I dedicate this section of my work.

I am profoundly grateful to my supervisor, Dr. Marieke Mur, for her invaluable mentorship and insights. Her expertise in addressing complex academic research questions and her generous provision of opportunities for personal and professional growth have been monumental in my journey. Working under her guidance has been both an honor and a transformative experience.

In times when distance from home was felt most acutely, the Heil family extended a kindness and warmth. This gesture of kindness will forever hold a special place in my heart. I owe a special debt of gratitude to Dalton Heil, whose unwavering support and insightful advice were instrumental in the completion of my dissertation. Dalton’s steadfast belief in my potential, even in moments of self-doubt, has been a source of immense encouragement.

I also wish to extend my profound appreciation to Ehsan Tousi. His role as a guiding voice in the development of my manuscript, along with his consistent and patient guidance, has been instrumental in equipping me with essential academic skills.

Lastly, I must express my special thanks to Brian Corneil, whose assistance was key in facilitating a master’s program that allowed me to simultaneously work at Meta FAIR and complete my thesis. His support and facilitation have been crucial in bridging my academic pursuits with real-world applications, enriching my educational experience significantly.
Acknowledgements

We extend gratitude to Farzad Shayanfar for his exceptional help in leading initial efforts in the development of the stimulus generation pipeline. His expertise and contributions have been invaluable to the success of this project.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Lay Summary</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>List of Abbreviations</td>
<td>xiv</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Motivation | 1

1.2 Human Object Recognition | 4

1.2.1 Ventral Visual Pathway | 4

1.2.2 Invariant Object Recognition | 7

1.2.3 Developing Invariant Object Recognition | 9

1.3 Computational Models of Human Object Recognition | 10

1.3.1 Modeling the Ventral Visual Pathway | 10

1.3.2 Neural Network Architectures | 10

1.3.3 Activation Functions | 14

1.3.4 Learning in Deep Neural Networks | 28

1.3.5 Invariant Object Recognition | 35

1.4 Thesis Objectives | 37
2 Methods

2.1 Stimuli

2.1.1 Category Mapping for Fair Comparison

2.1.2 3D Object Models

2.1.3 3D Transformations & Stimulus Generation

2.2 Experimental Design

2.2.1 Participants

2.2.2 Human Experiment

2.2.3 Neural Network Experiments

2.2.4 Error Consistency

3 Results

3.1 Invariant Object Recognition in Humans and Computational Models

3.2 Computational Models Limitations vs Humans

3.3 Effect of Architectural Base and Model Complexity on Performance

3.4 Effect of Model Visual Diet on Performance

3.5 Effect of Learning Objective on Performance

3.6 Error Consistency between DNNs and Humans

3.7 Explained Variance in Model Performance through Regression Analysis

4 Discussion

4.1 DNNs are sensitive to 3D object transformations

4.2 Humans are more invariant than DNNs to object rotation in depth

4.3 Humans and DNNs make different errors

4.4 Visual diet and learning objectives are important for developing invariant object recognition

4.5 Limitations

4.6 Future work
List of Figures

1.1 Perceptron Architecture. The computation performed by each perceptron involves summing the products of inputs \( x_i \) and their corresponding weights \( w_i \), followed by the application of a nonlinear function to the result \( \alpha \). ... 12

1.2 Multilayer Perceptron Architecture. Also called deep feedforward network or feedforward neural network. ... 13

1.3 Computation of a convolutional layer. On the left, example of 1D convolution operation. On the right, example of 2D convolution operation. ... 17

1.4 Computation of a max pooling layer. On the left, example of 1D max pooling operation. On the right, example of 2D max pooling operation. ... 18

1.8 ViT block overview. ViT block contains two main components: a self-attention mechanism and an MLP. The structure is designed to capture the interactions among input elements. ... 26

1.9 Auto-Encoders Architecture. composed of two sub-networks i.e. the encoder and the decoder. ... 31

2.1 An overview of how 3D object models were sourced, pre-processed, and used to generate images for evaluating humans and computational models. ... 41
2.2 Overview of the selected 3D object models. The objects are ordered, such that columns reflect categories and rows reflect exemplars. These objects were used in experimental trials.

2.3 Overview of the 3D object models that were used in practice trials.

2.4 Schematic of the TDW space, showing object and camera location, as well as images generated by the camera before object rotation (reference) and after object rotation about each axis.

2.5 An overview of all transformations used to generate images. The black border highlights object rotation in depth about the x axis, the transformation that we used for the human experiment.

2.6 An overview of the human experiment. Participants were instructed to categorize object images in a forced-choice paradigm. (a) Each trial started with a fixation cross that participants to click on, followed by image and mask presentation for 200 ms each, and a response screen that was presented for 1500 ms or until the participant clicked on one of the category icons, whichever came earlier. Images and masks were presented at 5 degrees of visual angle. (b) Schematic of the experiment, which consisted of three practice blocks and 10 experimental task blocks. Unknown to the participants, the second practice block served as a test: only participants with an accuracy of 80 percent or higher were included in the experiment and invited to continue.

2.7 An overview of all design features that DNNs were varied across: architecture, learning objective, and visual diet. Architectural changes can be varied either through architectural base (convolutional vs vision transformer) or model complexity (small, base, large).
3.1 An overview of the performance of more than 60 models and human participants concerning the 3D transformation involving object rotation along the x-axis. The background bars in grey illustrate the baseline performance of models when presented with canonical views. The dark foreground bars in the foreground represent the average performance across various transformations. The text placed below the bars provides the names of the corresponding models. The red bar highlights the average performance achieved by human participants. The dashed line signifies change level accuracy.

3.2 Average performance across all models on all 3D transformation. The dashed line represents the accuracy achieved on the canonical view, offering a baseline reference for comparison.

3.3 Average performance gap (i.e. performance for canonical - transformed object views) for models (x axis) and humans (y axis) across 16 categories.

3.4 A summary of human participants’ performance (a) and reaction time (b) when encountering the 3D transformation involving object rotation along the x-axis. The bars represent performance/reaction time across (1) all categories, (2) the category with the lowest performance/reaction time, and (3) the category with the highest performance/reaction time. The dashed line signifies accuracy/reaction time achieved on the reference view for all categories, providing a benchmark for comparison.
3.5 Summary of 8 models’ and humans’ performance on object rotation along the x-axis 3D transformations, highlighting the interplay between architecture and model complexity. Three architectures, namely Conv, ViT, and ConViT, are presented, each with three levels of model complexity: Small (S), Big (B), and Large (L). The background grey bars denote baseline performance on canonical views, while foreground colored bars depict average performance across all object rotations on the x-axis 3D transformations. Different colors correspond to specific model architectures, except the red bar which displays average participants’ performance.

3.6 Summary of 8 models’ performance on four 3D transformations (object scale, object rotation on x, y, and z axes) across 8 degrees of variation, highlighting the interplay between model complexity and architecture. Three architectures, namely Conv, ViT, and ConViT, are presented, each with three levels of model complexity: Small (S), Big (B), and Large (L). The dashed line at the bottom denotes the performance at chance level, while the two dashed lines at the top of each sub-figure represent the range of performance observed on reference views.

3.7 Summary of 6 models’ and humans’ performance on object rotation along the x-axis in 3D transformations, highlighting the effect of visual diet on performance. One architecture, namely ViT, is presented, with three levels of model complexity: Small (S), Big (B), and Large (L) and either trained with 1.28 million vs 14 million images. The background grey bars denote baseline performance on canonical views, while foreground colored bars depict average performance across all object rotations on the x-axis 3D transformation. Different colors correspond to specific model architectures, except the red bar which displays average participants’ performance.
3.8 Summary of 6 models’ performance on four 3D transformations (object scale, object rotation on x, y, and z axes) across 8 degrees of variations, highlighting the interplay between model complexity and architecture. One architecture, namely ViT is presented, each corresponding with three levels of model complexity: Small (S), Big (B), and Large (L) and either trained with 1.28 million vs 14 million images. The dashed line at the bottom denotes the performance at chance level, while the two dashed lines at the top of each sub-figure represent the range of performance observed on reference views.

3.9 Summary of 6 models’ and humans’ performance on object rotation along the x-axis in 3D transformations, highlighting the effect of learning objective on performance. One architecture, namely ViT, is presented, with three levels of model complexity: Small (S), Big (B), and Large (L) and trained with supervised, semi-supervised, self-supervised learning objective. The background grey bars denote baseline performance on canonical views, while foreground colored bars depict average performance across all object rotation on the x-axis 3D transformation. Different colors correspond to specific model architectures, except the red bar which displays average participants’ performance.

3.10 Summary of 6 models’ performance on four 3D transformations (object scale, object rotation on x, y, and z axes) across 8 degrees of variations, highlighting the interplay between model complexity and learning objective. One architecture, namely ViT is presented, each corresponding with three levels of model complexity: Small (S), Big (B), and Large (L) and trained with supervised, semi-supervised, self-supervised learning objective. The dashed line at the bottom denotes the performance at chance level, while the two dashed lines at the top of each sub-figure represent the range of performance observed across all human participants on reference views.
3.11 Analysis of error consistency for in-depth object rotation, revealing a gradient from low consistent errors (depicted in lighter colors) between human participants and models, to moderately consistent errors among human participants, to highly consistent errors (depicted in red) among DNN models. The similarity in error consistency patterns among DNNs illustrates the shared challenges they face in recognizing objects under varying degrees of in-depth rotation.

3.12 Averaged confusion matrices across models (left) and human participants (right) for 17 categories, the last category being 'none'. 'None' only used in human experiments for cases where participants did not provide a response. Colorbar indicates the number of conditions (80 conditions per category).

3.13 Analysis of explained variance in model performance achieved through linear regression, utilizing model architecture (base and complexity), visual diet, and learning objective as predictor variables. Architectural base reflects whether a DNN is a convolutional or a vision transformer network; model complexity reflects the size of the network; visual diet reflects the number of images in the training set; and learning objective reflects whether the network was trained using a supervised, semi-supervised, or self-supervised learning objective. The full regression model contains all four factors. The plot showcases the extent to which the factors contribute to the variability observed in the performance of different models across a range of tasks and transformations. Asterisks indicate regression models that explained significantly more variance than a regression model with a constant term only.
List of Abbreviations

- **CNN** Convolutional Neural Network
- **DNN** Deep Neural Network
- **SOTA** State-of-the-art
- **BN** Batch Normalization
- **ReLU** Rectified Linear Units
- **MLP** Multi-layer Perceptron
- **MAE** Masked AutoEncoders
- **ODD** Out-of-distribution
- **ResNet** Residual Neural Network
- **SOTA** State-of-the-art
- **ViT** Vision Transformer
- **IT** Inferotemporal Cortex
- **PPA** Parahippocampal Place Area
- **FFA** Fusiform Face Area
- **EBA** Extrastriate Body Area
- **LOC** Lateral Occipital Complex
Chapter 1

Introduction

1.1 Motivation

Humans have a remarkable ability to capture information about the surrounding environment through visual object recognition. The faculty of visual object recognition in humans stands out as a testament to our evolved cognitive capabilities. Understanding how the brain accomplishes the seemingly simple act of attributing an observed image to a pre-established category, such as discerning an apple from a car or a horse, has been a major interdisciplinary challenge. Object recognition, while on the surface appearing instantaneous and straightforward, draws upon a complex series of neural processes that transform fleeting visual stimuli into tangible, coherent mental constructs (Marr [2010]). Decades of cognitive neuroscience research have demonstrated that the brain accomplishes visual object recognition through a cascade of hierarchically organized brain areas referred to as the ventral visual pathway (Ungerleider & Mishkin [1982]; Goodale & Milner [1992]). Visual signals are transformed into meaningful object representations when moving along the pathway from early visual areas to inferior temporal areas.

Recent work suggests that it is possible to model the visual cortex using deep neural networks (DNNs) (Khaligh-Razavi & Kriegeskorte [2014]; Al-Tahan & Mohsenzadeh [2021]).
These computational models draw inspiration from the primate visual system to execute real-world object recognition tasks. While DNNs aren’t directly designed to mimic brain information processing but rather to complete a behavioral task, DNNs have been found to predict brain activity throughout the primate visual system on object recognition tasks (Khaligh-Razavi & Kriegeskorte, 2014; Yamins et al., 2014; Güçlü & Gerven, 2015). These models also have been shown to surpass the capabilities of earlier computational models of the visual system (Khaligh-Razavi & Kriegeskorte, 2014; Yamins et al., 2014). However, DNNs still leave substantial amounts of variance in brain representations unexplained (Schrimpf et al., 2018; Mell, St-Yves, & Naselaris, 2021; Storrs, Anderson, & Fleming, 2021) and struggle to model human behaviour on more challenging, naturalistic visual tasks such as recognizing objects in noisy visual input (Geirhos et al., 2018).

Remarkably, the ventral visual pathway exhibits a robust ability to identify objects irrespective of their orientations, placements, and sizes (Biederman, 1987; DiCarlo & Cox, 2007; Eger, Ashburner, Haynes, Dolan, & Rees, 2008; Freiwald & Tsao, 2010). This ability has been referred to as invariant object recognition. Any comprehensive model aiming to replicate human object recognition should emulate this human capability, demonstrating proficiency in recognizing objects even when they undergo various real-world transformations (DiCarlo, Zoccolan, & Rust, 2012). In essence, a model’s effectiveness should be gauged by its adaptability and versatility in handling diverse visual scenarios, akin to human visual cognition (Peters & Kriegeskorte, 2021; Bowers et al., 2022). Recent work in computer vision has begun to examine robustness of DNNs to variations in object orientation, particularly through the use of 3D graphics renderers (Madan et al., 2022; Alcorn et al., 2019; Abbas & Deny, 2022). However, there remains a noticeable void in the literature when it comes to comparing the performance of these networks to human capabilities. This leaves open the following questions: Are DNNs as invariant to variations in object orientation as humans are? Which design features make them most invariant?

We will address these questions by developing a stimulus set that systematically varies
object orientation, and that enables direct comparison between humans and DNNs on object recognition performance. By comparing performance between humans and DNNs on the same task, we aim to reveal to which extent DNNs can model human invariant object recognition. Furthermore, by testing a diverse range of DNNs with different design features, we will explore what makes some DNNs display more invariant object recognition behaviour than others. Their inherent flexibility and adaptability enable us to methodically adjust certain parameters, thus affording us the singular capability to individually analyze the role of visual diet, learning objectives, and inductive biases (Smith, Jayaraman, Clerkin, & Yu, 2018; Konkle & Alvarez, 2022; Goyal & Bengio, 2022). This approach provides a rare chance to probe the nuanced ways in which each of these elements influences the process of invariant object recognition.

We hope that the work presented in this thesis shows the potential of an integrated empirical-computational approach to understanding human vision. The interplay between computational models and the understanding of the brain is mutually beneficial (Hassabis, Kumaran, Summerfield, & Botvinick, 2017; Richards et al., 2019). On the one hand, computational models offer predictive capabilities that can shed light on potential underlying brain mechanisms. This allows us to systematically explore, integrate, and generate hypotheses about brain functions across different scales (Doerig et al., 2023). When such predictions are corroborated by empirical evidence, they augment our grasp of the underlying neural mechanisms. Conversely, the intricate architecture and operations of the brain have been foundational in the design and evolution of computational models (Hassabis et al., 2017). DNNs are a testament to this, being inspired by the brain’s neuronal interconnections (Lecun, Bottou, Bengio, & Haffner, 1998; Krizhevsky, Sutskever, & Hinton, 2012). Furthermore, the brain’s unparalleled efficiency, resilience, and sophisticated learning mechanisms can guide the creation of more adept and energy-efficient computational methods (Spöerer, Kietzmann, Mehler, Charest, & Kriegeskorte, 2020). Thus, as we endeavor to unravel the mysteries of the brain, we not only gain profound insights into our own existence but also fuel advancements in computational modeling, thereby catalyzing progress in both domains. Lastly, the understanding of ventral visual pathway and its role in visual recognition not only sheds light on the marvels of human cognition but also sets
the stage for exploring potential applications and interventions in fields ranging from artificial intelligence to clinical neurology.

1.2 Human Object Recognition

1.2.1 Ventral Visual Pathway

Human dependency on visual cues is pivotal for understanding and interacting with the world around us. Everyday activities, like finding our car keys, safely navigating through bustling streets, or spotting familiar faces in a crowd, hinge on our ability to perceive and recognize objects. Though recognizing objects might seem straightforward, it’s computationally complex (DiCarlo & Cox, 2007). We’re required to identify objects under varying conditions, such as different angles, light levels, or their relative positioning. Moreover, we group objects based on their functional significance, even if they don’t look identical (Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976). This means categorizing objects into broad, behaviorally meaningful groups, like living entities or faces. Such categorization streamlines our reactions to the objects we come across, enabling us to react aptly even to objects we’ve never seen before.

The very foundation of our visual experience begins in the retina with rod and cone photoreceptors. These specialized cells transduce ambient light signals into electrical responses, paving the way for the intricate cascade of processes that comprise vision (Goldstein, 2022). Rods, abundant in the peripheral retina, are sensitive to low light conditions and primarily contribute to our night vision. In contrast, cones, predominantly situated in the central retina (particularly the fovea), are responsible for our color perception and detailed vision in brighter conditions. Once these photoreceptors have converted light into electrical signals, the information is processed further in the retina and then transmitted to the brain through the optic nerve (Goldstein, 2022). Herein begins the journey of visual information through the visual pathway,
one of the brain’s most sophisticated and vital subsystems dedicated to visual object recognition.

Central to the visual pathway, numerous studies on monkeys with lesions (Ungerleider & Mishkin 1982) and humans (Goodale & Milner 1992) demonstrated that there are two primary streams that serve distinct but interrelated functions in processing visual information: the ventral and dorsal pathways. The ventral pathway, often called the "what" pathway, starts from the primary visual cortex (V1) and extends towards the inferotemporal (IT) cortex in the brain’s anterior portions. The ventral pathway is responsible for object recognition. It allows us to identify and categorize visual stimuli based on their characteristics, such as color, shape, size, and texture (Ungerleider & Mishkin 1982; Tanaka 1996). This pathway helps us distinguish between a cat and a dog or recognize a familiar face in a crowd. The dorsal pathway, colloquially termed the "where" pathway or sometimes the "how" pathway, originates from V1 and leads towards the posterior parietal cortex. This pathway is paramount for processing spatial information, including the location of objects in our visual field and their spatial relationship to each other (Goodale & Milner 1992). Furthermore, it plays a role in guiding our movements in relation to these objects. For instance, when reaching out to grab a cup, it’s the dorsal pathway that informs the brain where the cup is in relation to our hand and how to move the hand to grasp it (Marr 2010; Mishkin, Ungerleider, & Macko 1983). The focus of the thesis will be on understanding the "what" component in a visual scene, hence will primarily investigate the ventral visual pathway.

At the core of the visual processing hierarchy in the brain lies V1. Located in the occipital lobe, V1 stands as the initial cortical processing stage for visual information relayed from the retina through the lateral geniculate nucleus of the thalamus (Hubel & Wiesel 1968). Early work on V1 in cats demonstrated that particular neurons in this area optimally react to edges with distinct orientations (Hubel & Wiesel 1959). This observation suggested that V1 handles basic visual elements, such as edges and angles.

Beyond mere orientation, V1 has also been shown to play a crucial role in processing
spatial frequency, contrast sensitivity, and color through its diverse array of neurons (Livingstone & Hubel, 1988). Additionally, the retinotopic structure in V1, characterized by adjacent cortical cells reacting to adjacent sections of the visual field, guarantees a consistent and structured representation of the visual environment (Tootell et al., 1998). However, V1’s contributions are not isolated to basic feature detection. V1 is also implicated in more complex processes. Lamme and Roelfsema (2000) demonstrated that neurons in V1 exhibit sensitivity to the broader organization of the visual scene, likely stemming from feedback processing mechanisms and lateral connections from pyramidal neurons. For example, feedback from areas like V2, V4, or IT, which possess larger receptive fields, can reshape V1’s reactions, leading to effects that account for context or extend beyond the classical receptive field (Sillito, Cudeiro, & Jones, 2006).

Immediately following V1 is the secondary visual cortex, V2, which provides an essential link in the chain of visual information processing. The functional properties of single V2 cells could be aptly predicted by aggregating responses from multiple V1 cells. This aggregation results in a slight expansion in receptive field size in V2 cells and an enhanced selectivity for more intricate stimuli, such as angles, distinguishing it from the simpler orientation selectivity characterizing V1 (Hegdé & Van Essen, 2000). Such findings have further fortified the understanding that the ventral visual pathway operates in a serial hierarchical manner (Riesenhuber & Poggio, 2000). With each subsequent stage in this hierarchy, there’s an augmentation in the complexity of selectivity, coupled with a growing invariance to basic visual alterations like retinotopic position. In essence, as visual information journeys from V1 to V2 and beyond, it undergoes sophisticated transformations, refining raw visual data into more contextually nuanced and globally relevant interpretations.

Bridging the early visual processing regions to IT cortex is V4, which is more deeply implicated in fine-tuning representations suited for object recognition. V4 has been shown to parse objects from visual scenes, acting as an interpreter for encountered visual cues (Kim, Bair, & Pasupathy, 2019). Visual representations are tuned to form, color, depth, texture,
and other complex attributes such as simple shapes (Pasupathy & Connor, 2001), crafting high-dimensional representations that are vital to distinguishing one object from another.

The inferotemporal (IT) cortex stands out as a hub for high-level visual processing, wherein neurons exhibit a heightened specificity for complex visual features (Tanaka, 1996). One of the striking characteristics of neurons within the IT cortex is their selectivity for particular visual stimuli. For instance, many IT neurons demonstrate a penchant for distinct object shapes, with certain neurons even displaying a predilection for specific objects or faces (Gross, Rocha-Miranda, & Bender, 1972; Tanaka, 1996). Several specialized sub-regions within IT have been identified, each catering to the recognition of specific categories of visual stimuli. The fusiform face area (FFA) plays a pivotal role in facial recognition and is found to be more active when individuals view faces compared to other objects (Kanwisher, McDermott, & Chun, 1997; McCarthy, Puce, Gore, & Allison, 1997; Halgren, Raij, Marinkovic, Jousmäki, & Hari, 2000; Sergent, Ohta, & MacDonald, 1992). In a parallel vein, the parahippocampal place area (PPA) is predominantly activated when viewing scenes or places, highlighting its significance in spatial recognition (Epstein & Kanwisher, 1998). Complementing these are regions like the extrastriate body area (EBA) that is implicated in the perception of body parts other than faces (Downing, Jiang, Shuman, & Kanwisher, 2001), and the lateral occipital complex (LOC) which is crucial for object recognition, demonstrating selectivity for the shape or form of objects (Malach et al., 1995). The complexity of these visual representations implies that IT neurons integrate information over larger portions of the visual field compared to neurons in earlier visual areas.

### 1.2.2 Invariant Object Recognition

Invariant object recognition, or the ability to robustly identify objects irrespective of their orientations, placements, and sizes, is a major computational challenge for the human ventral visual pathway (Biederman, 1987; Riesenhuber & Poggio, 2000; DiCarlo & Cox, 2007). Changes in an object’s orientation, placement, or size are often referred to as identity-preserving
Transformations: they do not change the object’s identity, but they can have a large effect on the visual appearance of an object. This is especially the case for changes in object orientation. For example, two different faces seen in the same orientation (e.g. front view) are more similar to each other in terms of low-level visual properties than the same face seen in different orientations (e.g. front view and profile view). To successfully recognize objects across identity-preserving transformations, the ventral visual pathway needs to abstract from changes in visual appearance associated with these transformations (DiCarlo & Cox, 2007). At the same time, it needs to be sensitive to visual differences associated with object identity or category membership (Konkle & Alvarez, 2022; Grill-Spector & Weiner, 2014).

Despite the computational challenges associated with recognizing objects across identity-preserving transformations, behavioral and neuroscience evidence indicates that humans are good at this task (Biederman, 1987; Eger et al., 2008; Isik, Meyers, Leibo, & Poggio, 2014; Kar, Kubilius, Schmidt, Issa, & DiCarlo, 2019). Several theories have been proposed to account for these findings. One prominent early proposal is "recognition by components" which posits that objects are collections of 3D components and predicts that we can recognize objects from different viewpoints as long as we can see enough of the object’s components in their known spatial arrangement (Biederman, 1987). This proposal echoes David Marr’s notion of an object-centered representation (Marr, 2010). A second major proposal, which gained popularity slightly later, is that we recognize objects from different viewpoints by integrating information across multiple (known) 2D object views (Bülthoff, Edelman, & Tarr, 1995; Riesenhuber & Poggio, 2000). An advantage of the second proposal is that it is associated with explicit computational models of the ventral visual pathway. This provides a complementary avenue for testing mechanisms of invariant object recognition, in addition to testing predictions on human data. DNNs can be considered extensions of the early view-based models; they use the same hierarchical feedforward architecture and convert 2D object views into a meaningful description of the object (Krizhevsky et al., 2012).
1.2.3 Developing Invariant Object Recognition

In humans, invariant object recognition emerges over the course of development, starting within the first few months of life (A. J. Caron, Caron, & Carlson, 1979; Day & McKenzie, 1981; Gliga & Dehaene-Lambertz, 2007). Robustness to changes in object placement and size appear to develop faster than robustness to changes in object orientation (Gliga & Dehaene-Lambertz, 2007; Nishimura, Scherf, Zachariou, Tarr, & Behrmann, 2015), suggesting that the latter may be more computationally challenging. We only touch briefly here on the developmental trajectory of invariant object recognition in humans, because the focus of this thesis is on contrasting and comparing invariant object recognition between human adults and pre-trained ("adult") DNNs. Nevertheless, in this endeavor, we will use DNNs as a modeling platform for testing several factors hypothesized to contribute to the development of invariant object recognition in humans.

These factors include (1) inductive biases built into the visual system at birth, including the convolutional and hierarchical architecture of the ventral visual pathway (Ellis et al., 2021; Arcaro & Livingstone, 2017), which may facilitate the development of invariance to changes in an object’s placement within the visual field; (2) the visual input that humans receive early in life, also referred to as an infant’s visual diet, which has been shown to sample only a small number of objects but across a wide range of identity-preserving transformations (Smith et al., 2018); and (3) behavioral pressures such as the need to discriminate each object from each other object despite changes in visual appearance due to identity-preserving transformations (Konkle & Alvarez, 2022). In the computer vision literature, the third factor is often referred to as the learning objectives of the individual or system (Richards et al., 2019). The above factors may not be completely independent in the real world. For example, an infant may selectively attend to visual inputs that are relevant for achieving its learning objectives, and as such curate its visual diet.
1.3 Computational Models of Human Object Recognition

1.3.1 Modeling the Ventral Visual Pathway

Since the visual neuroscience discoveries by Hubel and Wiesel in the 1960s, researchers in both neuroscience and computer science have endeavored to model (components of) the ventral visual pathway. In neuroscience, early work focused on modeling responses of V1 neurons with Gabor filters (Jones & Palmer, 1987; Olshausen & Field, 1996). This approach was successful, but it proved challenging to build filters that simulated the more complex response properties of neurons in V2 and beyond. Until a decade ago, responses in IT cortex were best modeled using semantic descriptions of visual stimuli (Kriegeskorte et al., 2008; Huth, Nishimoto, Vu, & Gallant, 2012). In computer science, work focused on connectionist approaches for visual pattern discrimination (Rosenblatt, 1958; Fukushima, 1980; Lecun et al., 1998), which for long did not scale up to real-world object recognition problems (Serre, Oliva, & Poggio, 2007). However, thanks to advances in parallel computing and the availability of large amounts of training data, these models have begun to fulfill their potential as system-level models of the ventral visual pathway (Krizhevsky et al., 2012). A benefit of these neural network models is that researchers do not need to handcraft filters: the models learn filters from visual input given a certain learning objective. The following paragraphs will discuss neural networks in more detail.

1.3.2 Neural Network Architectures

Perceptron

Neural networks, loosely inspired by biological neural networks in the brain, form a family of computational models designed to uncover underlying relationships within a dataset. These
relationships are discovered through a collection of units known as nodes or neurons. In a neural network, each "neuron" is represented by a mathematical function that establishes connections with neurons in the preceding layer, using weights \((w_i)\) associated with inputs \((x_i)\). Each neuron takes input data in the form of feature values (e.g., pixel values in an image, numerical attributes in a dataset). Each feature is represented by a numerical value, and the neuron processes multiple input features simultaneously. For each input feature, the neuron assigns a weight, which reflects the feature’s importance or contribution to the decision-making process. The computation performed by each neuron involves summing the products of inputs \(x_i\) and their corresponding weights \(w_i\), followed by the application of an activation function \(\alpha\) to the result (Equation (1.1)). The activation function introduces non-linearity into the model which determines whether the neuron should be "activated" or "deactivated" based on the computed sum. A neural network with only one neuron and an input and output layer is referred to as a Perceptron (Figure 1.1):

\[
y = \alpha \left( \sum_{i=1}^{n} w_i \cdot x_i + b \right)
\] (1.1)

**Multi-layer Perceptron**

Perceptrons are limited to approximating linear functions, which resemble multiple linear regressions. To overcome this limitation, we can stack perceptrons vertically, creating a layer of interconnected perceptrons, also known as the model’s width. When these layers of perceptrons, referred to as hidden layers, are interconnected, we transition into a multi-layered perceptron (MLP), also called a deep feedforward neural network, capable of approximating non-linear functions and tackling significantly more intricate problems (Figure 1.2). The term feedforward originates from the fact that each layer’s output serves as the input for the succeeding layers, without any feedback connections to the current or previous layers. The learning algorithm updates the weights of the hidden layers iteratively until the neural network achieves minimal error. During training, the samples specify the desired output for a given input at the output layer,
Figure 1.1: Perceptron Architecture. The computation performed by each perceptron involves summing the products of inputs ($x_i$) and their corresponding weights ($w_i$), followed by the application of a nonlinear function to the result ($\alpha$).

MLPs demonstrate computational efficiency when dealing with relatively low-dimensional input spaces. However, as the input size increases, MLPs become less practical for such problems. Consider the scenario where we have an image dataset composed of colored images in $\mathbb{R}^{3\times m\times n}$, and we want to employ a model to predict the classes of these images. Flattening the 2-dimensional image into a vector in $\mathbb{R}^{3mn}$ and feeding it through an MLP, with each pixel as a feature, would lead to a significant escalation in the number of parameters as a function without imposing constraints on the hidden layers. As the information progresses through the feedforward connections, the weights learn representations based on the output of the preceding layer, thus progressively influencing the neural network’s subsequent layers.
1.3. **Computational Models of Human Object Recognition**

Figure 1.2: Multilayer Perceptron Architecture. Also called deep feedforward network or feedforward neural network.

... of $m$ and $n$, making the model difficult to train. Moreover, when dealing with 2-dimensional (or n-dimensional) data, we lose spatial information, that is, spatially closer pixels possess spatial coherence, a characteristic that a simple feedforward neural network fails to capture. These limitations are among the primary reasons why current state-of-the-art architectures predominantly rely on Convolutional Neural Networks (Lecun et al., 1998) (CNNs) to extract feature maps from high-dimensional data, such as images, audio, and video. CNNs are better suited to handle the spatial relationships within the data, making them the preferred choice for tasks involving complex, high-dimensional data.
1.3.3 Activation Functions

In the preceding section, we introduced activation functions, but we did not delve into the various types and the reasons behind their varying usage. When a hidden unit in a neural network computes a weighted sum of its inputs, the resulting output can range from $-\infty$ to $\infty$, making it challenging to convey to subsequent hidden units whether a particular unit has "activated" or not. Activation functions serve the purpose of non-linearly bounding the outputs, conveying vital information about the hidden unit’s state. These functions must be non-linear, as a neural network comprising only linear activation functions across multiple layers would reduce to a mere linear function of its first layer’s input. In this section, we will explore some of these activation functions:

1. **Sigmoid Function**: transforms the input into a value between $[0, 1]$, closely approximating one for large inputs and approximating zero for small inputs. The sigmoid function’s bounded output prevents exploding activations. On the other hand, the values on either sides tend to respond very less to changes in the input, hence the gradient at that region is going to be small (i.e. vanishing gradient).

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$  \hfill (1.2)

2. **Hyperbolic Tangent (tanh)**: is very similar to the sigmoid function, in fact, $tanh(x) = 2\sigma(2x) - 1$. One distinction is that $tanh$ gradient is stronger in magnitude, with the bounded range of the activations are between $[-1, 1]$. However, this also mean that similar to the sigmoid function, $tanh$ is still susceptible to vanishing gradient problem.

$$tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$  \hfill (1.3)

3. **Rectified Linear Units (ReLUs)**: simply output the input if the input is positive, otherwise, it outputs 0. This means that the function is linear for values greater than zero. While
1.3. **Computational Models of Human Object Recognition**

the non-linear property of ReLU are in the negative values were negative values are always transformed to zero. These characteristics makes ReLU an attractive activation function for various reasons: (1) *computational cost*: ReLU is less computationally expensive than tanh and sigmoid functions because it does not require the use of an exponential calculation. (2) *Sparsity*: unlike tanh and sigmoid functions, ReLU is capable of outputting a true zero value. Meaning that some hidden units in the network are not activated and thereby making the activations sparse and efficient. This is a desirable property as it can accelerate learning, simplify the model, and overcome vanishing gradients problem.

\[ ReLU(x) = \max\{0, x\} \quad (1.4) \]

One major drawback of using ReLU is that they cannot learn on examples for which the input activations are less than equal to zero. For activations in the negative region of ReLU, the gradient will be zero, hence, the weights will not be adjusted based on that input. This means that some neurons can potentially get stuck in a state were variations in the input always results in zero, often called *dying ReLU*. Various adjustments were proposed to ReLU to mitigate this problem. For instance, absolute value rectification (Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009), leaky ReLU (Maas, Hannun, & Ng, n.d.), or PReLU (??).

**Convolutional Neural Networks (CNNs)**

CNNs are exceptionally proficient at processing data with a known 2-D spatial structure. The name "Convolutional Neural Networks" is derived from the mathematical operation, known as *convolutions*. One of the early CNN architectures, LeNet-5 (Lecun et al., 1998), gained recognition for its successful implementation and training on the MNIST dataset. The core of CNNs involves the utilization of convolutional operations, predominantly in the early layers, where the input space retains high dimensions. However, as the input space undergoes pro-
gressive reduction, an MLP (Multi-Layer Perceptron) typically performs the final task, such as classification. In this section, we will explore the essential components that constitute a CNN:

1. **Convolutional Layers** (Figure 1.3): Convolutional Layers consist of three crucial components: the input, the kernel, and the feature maps (output). The kernel is often smaller in size than the input but possesses the same dimensionality. It "slides" or moves over the input, performing an element-wise multiplication with the part of the input it currently covers and then summing up the results into a single output. This process is repeated for every location the kernel "slides" over, effectively converting the input matrix of features into another matrix of features. The output features are the weighted sums of the input features located approximately in the same position as the output pixel on the input layer. Mathematically, this operation can be defined as follows for a 1D input vector:

\[
(f \ast g)(i) = \sum_{m} g(m) \cdot f(i - m)
\]  

(1.5)

where \( f \) is a 1D input vector, \( g \) is a 1D kernel vector, \( N \) is the length of \( f \), and \( M \) is the length of \( g \). We perform this operation for every \( i \in N \). This operation can be generalized to 2D by convolving over more than one axis at a time:

\[
(f \ast g)(i, j) = \sum_{m} \sum_{n} g(m, n) \cdot f(i - m, j - n)
\]

(1.6)

The amount of movement the kernel steps per computation is determined by the value of \( stride \) (default = 1).

2. **Pooling Layers** (Figure 1.4): neural networks are tasked with extracting essential information from high-dimensional, low-level sensory data (such as images or audio). Convolutions play a crucial role in extracting these features. However, as we advance to higher levels of representation, it becomes necessary to reduce unnecessary spatial dependencies among features. This reduction is achieved through the use of pooling layers. The pooling layer performs down-sampling operations, effectively reducing the spatial dimensions of the generated feature maps. There are various types of down-sampling
1.3. **Computational Models of Human Object Recognition**

Figure 1.3: Computation of a convolutional layer. On the left, example of 1D convolution operation. On the right, example of 2D convolution operation.

Operations, with maximum pooling being the most widely adopted in the literature. Maximum pooling calculates the maximum (i.e., the largest) value within each patch of the feature map, resulting in pooled feature maps that emphasize the most dominant features in each patch. Alternatively, average pooling involves calculating the average value within each patch of the feature map. Both pooling methods contribute to feature reduction and simplification, enabling the subsequent layers of the neural network to focus on the most salient and task-relevant features.

Other types of pooling methods, known as *global pooling*, are utilized on the output of the last convolutional layer. Unlike traditional pooling methods that down-sample patches of the input feature map, global pooling down-samples the entire feature map to a single value. This aggressive summarization of feature presence allows for a smooth transition...
Figure 1.4: Computation of a max pooling layer. On the left, example of 1D max pooling operation. On the right, example of 2D max pooling operation.

from feature maps to an output that is suitable for making predictions.

3. **Fully Connected Layers**: another crucial component of a CNN is the inclusion of fully connected layers, typically implemented as an MLP (Multi-Layer Perceptron). These layers receive the flattened or globally pooled output of the last convolutional layer as input. The output of the last convolutional layer consists of high-level features in a low-dimensional space. By adding fully connected layers, we efficiently learn non-linear combinations of these features. In classification tasks, setting the size of the last fully connected layer to match the number of classes enables each node to output the probability that the input belongs to a specific class. This facilitates the final classification of the input data. By incorporating fully connected layers, CNNs can effectively leverage the extracted high-level features to make accurate predictions and perform complex tasks.

Figure 1.5 depicts a basic CNN architecture using a sample image from the frog class in the CIFAR10 dataset (Krizhevsky 2009). The CIFAR-10 dataset comprises 60,000 32x32 colored images, categorized into 10 classes, with 6,000 images per class. Each convolutional block in Figure 1.5 consists of a convolution layer, max pooling layer, and ReLU activation function. While this architecture may suffice for the CIFAR10 dataset, over the past decade,
1.3. Computational Models of Human Object Recognition

CNNs have evolved to address more challenging datasets, particularly ImageNet (Deng et al., 2009) in the domain of Computer Vision. ImageNet, consists of an extensive collection of 1.2 million hand-labeled natural images spanning 1000 categories. The evolution of CNNs began with AlexNet (Krizhevsky et al., 2012), which achieved a breakthrough by employing 5 convolutional layers to surpass the state-of-the-art on ImageNet. Subsequently, GoogleNet (Szegedy et al., 2014) and VGG (Simonyan & Zisserman, 2015) were introduced, boasting 19 and 22 layers, respectively, further pushing the boundaries of performance on ImageNet. These deeper architectures allowed CNNs to tackle complex visual recognition tasks with remarkable accuracy and have since become foundational models for various computer vision applications.

Figure 1.5: Example CNN. Each convolutional block consists of a convolution layer, max pooling layer, and ReLU activation function. The image is a sample belonging to the frog class from CIFAR10 (Krizhevsky, 2009) dataset.

**Residual Neural Network (ResNet)**

Although increasing the depth of networks improved performance on ImageNet, it also led to an increased challenge in training due to the issue of vanishing gradient. The vanishing gradient problem arises when the gradients in early layers approach zero. This is not a significant problem for shallow networks with only a few layers, but when deeper networks are employed, it can severely hamper the effectiveness of the training process. To address this problem, early solutions introduced an auxiliary loss in a middle layer to help maintain a reasonable gradient flow (Szegedy et al., 2014). However, a more direct and effective solution was introduced with
the ResNet Architecture (2,3). This architectural innovation revolutionized deep learning by introducing the concept of residual blocks, enabling the successful training of extremely deep networks. By introducing skip connections that allow the network to learn residual functions, the vanishing gradient problem was effectively mitigated, making it feasible to construct much deeper architectures that achieved state-of-the-art performance on various tasks.

ResNet derives its name from the innovative use of residual blocks within its architecture. In 4, the residual connection directly adds the value at the beginning of the block to the end of the block, leading to the following equation:

\[ y = F(x) + x \quad (1.7) \]

where \( x \) is the input, \( F \) is the convolutions operating on the input. By adding the identity value of \( x \) through the residual connection, the input would skip an activation function that could vanish the gradient. This approach ensures that stacking layers does not degrade the network’s performance, as we can simply add identity mappings to prevent the gradient from approaching zero. Consequently, deeper models should not yield higher training errors compared to shallower counterparts, as the stacked layers can more easily fit a residual mapping rather than directly fitting the desired underlying mapping. It is important to note that typical residual blocks do not alter the spatial resolution of the input. However, since down-sampling our representations is necessary (as discussed earlier), we incorporate a residual block capable of down-sampling the input \( x \) through both streams (Figure 1.6) after a series of \( n \) residual blocks lacking this capability.

Figure 1.6 presents examples of the ResNet18 and ResNet34 architectures, with the key difference being the number of residual connections without down-sampling capability. Since its initial introduction, the ResNet architecture has been extensively studied and adapted with various variations (He, Zhang, Ren, & Sun, 2016b; Xie, Girshick, Dollár, Tu, & He, 2017; Huang, Liu, van der Maaten, & Weinberger, 2018; Liu, Mao, et al., 2022; Tan & Le, 2021).
Simonyan & Zisserman [2015], Kolesnikov et al. [2020], Tan & Le [2020]. This architecture will play a pivotal role in the subsequent chapters, warranting its comprehensive introduction here.

Figure 1.6: Residual blocks architecture [21]. (A) Illustrates Residual Block, showcasing the flow of inputs through layers with the addition of a shortcut connection enabling the model to learn identity functions. (B) Illustrates Residual Block with down-sampling, this component decreases the spatial dimensions of the input through the block while increasing the number of channels. (C) An overview of the ResNet50 architecture, presenting a network of 50 layers deep with repetitive stacking of residual blocks.

In statistical analysis, the concept of normalization refers to the procedure of transforming disparate value ranges to a unified scale, and recalibrating a particular data distribution to align with a standard normal distribution, characterized by a mean of 0 and a standard deviation of 1. This procedure often finds application in deep learning, where input data is typically normalized by deducting the mean:

\[
\mu = \frac{1}{m} \sum_{i} x_i
\]

\[
x := x - \mu
\]  

(1.8)

Additionally, variance normalization is applied:
This is a beneficial process as it addresses the potential discrepancies in the value ranges of different features, which could adversely affect both the performance and the learning speed of the model, and in some cases, completely impede the model’s learning capabilities. Batch normalization (Ioffe & Szegedy, 2015) expands this idea by incorporating these operations not just on the input data, but within the intermediate layers of the network as well (as depicted in Figure 1.6).

In a general sense, every layer within a deep neural network seeks to align the input data with certain representational patterns. However, as data travels through these layers, the occurrence of distributional shifts can pose challenges to the subsequent layers in acquiring meaningful representations, given the need to compensate for these drifts. Another crucial issue addressed by batch normalization is the problem of exploding and vanishing gradients. Certain activation functions can end up in their saturation zones, thereby rendering some representations immutable.

Batch normalization has indeed gained immense popularity in its application; however, a variety of other normalization methods have also been proposed, including Layer Normalization (Ba, Kiros, & Hinton, 2016), Instance Normalization (Ulyanov, Vedaldi, & Lempitsky, 2016), Group Normalization (Wu & He, 2018), and Weight Normalization (Salimans & Kingma, 2016). Each of these techniques presents its own advantages and disadvantages. For example, Layer Normalization is a method designed to standardize the inputs feeding into a neural network layer. Unlike Batch Normalization, which operates over the batch dimension, Layer Normalization takes a different approach by working across the feature dimension. This means that for each sample within a batch, Layer Normalization calculates the mean and standard deviation for all its features (dimensions) individually, subsequently applying this calculated normalization

\[ \sigma^2 = \frac{1}{m} \sum_{i}^{m} (x_i - \mu)^2 \]  

\[ x := \frac{x}{\sigma^2} \]  

(1.9)
1.3. **Computational Models of Human Object Recognition**

independently to each sample. Which allows the model to handle sequences of various lengths. It doesn’t require computing statistics across the batch dimension, which can vary with different sequence lengths.

Require: Values of $x$ over a mini-batch: $\{x_{1...m}\}$; 
Parameters to be learned: $\gamma, \beta$

Ensure: $\{y_i\}$

\[
\mu \leftarrow \frac{1}{m} \sum_{i} x_i \quad \text{mini-batch mean}
\]

\[
\sigma^2 \leftarrow \frac{1}{m} \sum_{i} (x_i - \mu)^2 \quad \text{mini-batch variance}
\]

\[
i \leftarrow \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad \text{normalize}
\]

\[
y_i \leftarrow \gamma_i + \beta \quad \text{scale and shift}
\]

Algorithm 1: Batch Normalizing Transform, applied to activation $x$ over a mini-batch (Ioffe & Szegedy, 2015).

Batch normalization, as defined in Algorithm 1 addresses these issues by applying normalization to the inputs of each intermediate layer based on the mean and standard deviation calculated over a specific mini-batch from the dataset. A notable differentiation between normalization of input data and batch normalization lies in the application of statistical measures. While input data normalization uses statistics derived from the whole dataset, batch normalization employs the mean and variance calculated over an individual mini-batch.

**Vision Transformers (ViT)**

ViTs have recently emerged as a competitive alternative to CNNs, with improved performance on image classification tasks and faster inference time (Dosovitskiy et al., 2021). Adopting architectures that were originally crafted for natural language processing (Vaswani et al., 2017),
ViTs repurpose these methods for image classification tasks. A pivotal component in the ViT architecture is the self-attention mechanism, also referred to as scaled dot-product attention or multi-head attention. This approach allows the model to discern and assign significance to different segments of the input sequence, thereby enabling the delineation of intricate relationships among image patches. In this segment, we will delve into the critical elements that empower ViT (Figure 1.7):

Figure 1.7: ViT overview. An image is split into fixed-size patches, linearly embed each of them, add position embeddings, and fed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, [Dosovitskiy et al. (2021)] added an extra learnable classification token to the sequence.

1. **Patch Embedding and Encoding**: The input image is divided into a grid of small, non-overlapping patches, each of a fixed size (e.g. 16x16 pixels). This operation effectively treats the image as a sequence of patches. Each patch is then flattened into a one-dimensional vector. This step transforms the two-dimensional patch into a format suitable
for processing by the subsequent transformer blocks. Following the flattening, a linear transformation is applied to each flattened patch to transform it into a patch embedding.

The transformer architecture doesn’t have a built-in sense of the relative positions of patches or the sequential order of the data it processes. Therefore, after the patch embedding step, positional information is added back to provide the model with some context about the location of patches within the image grid. A positional embedding (a learnable parameter of the same dimension as the patch embedding) is added to each patch embedding. These positional embeddings are learned during training and allow the model to encode the relative positions of patches in the image. The combination of patch embeddings and positional encodings serves as the input to the subsequent layers of the Vision Transformer. This design allows the model to consider both the content of each image patch and its relative position in the image.

2. **Scaled Dot-Product Attention**: provides a way for the model to focus on different parts of the input, paying "attention" to patches that are relevant to the task at hand. For each patch in the input sequence, three vectors are computed: the Query vector \( Q \), the Key vector \( K \), and the Value vector \( V \). These vectors are obtained by applying different learned linear transformations to the input patch embeddings. This mechanism allows each patch to consider all other patches when being processed, thus capturing the global context of the image. This is unlike traditional CNNs, which primarily operate on local receptive fields. This mechanism also allows the model to adaptively adjust the degree of attention it pays to different patches based on their importance for the task at hand. This operation can be summarized as the following:

\[
\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V
\]

(1.10)

where: \( Q, K, \) and \( V \) represent the query, key, and value matrices, respectively. Each of these matrices are derived from the input. \( d_k \) is the dimensionality of the queries and keys, which is used to scale the dot product to prevent it from growing too large as
the dimensionality increases. The softmax function is applied on the keys’ and queries’ dot product, ensuring the output values are between 0 and 1 and sum to 1. This makes these values interpretable as probabilities, representing the model’s attention scores. The attention scores are then used to weight the value vectors, which are subsequently summed to produce the output of the attention layer (Figure 1.8).

3. **Multi-Head Self-Attention**: is the core part of the transformer block where the model learns to pay different levels of attention to different patches of the image, depending on their relevance to the task at hand. Each 'head' in this layer is a scaled dot-product attention mechanism. These heads work in parallel and focus on different parts of the input sequence, allowing the model to capture various types of relationships among patches. The outputs of all the attention heads are concatenated and then linearly transformed into the input size for the next component.

![Figure 1.8: ViT block overview. ViT block contains two main components: a self-attention mechanism and an MLP. The structure is designed to capture the interactions among input elements.](image)

ViTs and CNNs are two different types of models used for image recognition tasks, and they differ in several fundamental ways:

1. **Architecture**: CNNs are designed based on the premise that local, spatial hierarchies exist within images. They employ local connections and parameter sharing through
the use of convolutional filters to extract these hierarchies, which are later aggregated through pooling operations to form a global image understanding. ViTs, on the other hand, leverage transformer architectures initially designed for natural language processing tasks. They treat an image as a sequence of patches, similar to how a sentence is treated as a sequence of words or tokens in NLP.

2. **Receptive Field:** The receptive field of CNNs is local and depends on the size of the convolutional kernel. On the other hand, transformers used in ViTs have a global receptive field due to the self-attention mechanism, which allows the model to consider dependencies between all patches in the image.

3. **Understanding of Positional Information:** CNNs inherently deal with positional information due to their convolutional and pooling operations. However, ViTs, after dividing the image into patches, lose this spatial context. To rectify this, ViTs incorporate position embeddings into the input sequence to provide the transformer with the positional context.

ViTs utilize a pair of factors that govern the intricacy of the model: the network’s depth and width, along with the quantity and dimensions of the image patches. Consequently, when referring to a ViT-B-32, it signifies that this is a specific configuration within the Vision Transformer architecture. In this context, "B" designates the foundational setup, dictating the depth and width of the neural network, while "32" denotes the dimensions of the patches used in processing the input images. It is crucial to note that ViT architecture has inspired several variants, each tailored to particular tasks or computational constraints (Touvron, Cord, & Jégou, 2022; Liu et al., 2021; Liu, Hu, et al., 2022).

**Convolutional Vision Transformers (ConViT)**

ConViT (d'Ascoli et al., 2021) integrates the strengths of both convolutional layers from CNNs and global attention from ViTs for image feature understanding. The main difference between a
standard ViT and ConViT lies in the formulation of the self-attention mechanism. The main difference between a standard ViT and a ConViT lies in the formulation of the self-attention mechanism. In the standard ViT, each token (which corresponds to a patch of an image) interacts equally with all other tokens in the sequence through the attention mechanism, making it a fully global attention method. The convolutional self-attention mechanism of the ConViT uses a fixed Gaussian-like kernel, which decays as the distance between tokens increases, hence favoring closer tokens in the attention operation. This kernel is designed to be broad enough to allow a good balance between local and global attentions.

1.3.4 Learning in Deep Neural Networks

Deep Neural Networks’ learning objectives are categorized based on the type of data they utilize during the training process: Supervised, Unsupervised, and Semi-Supervised. In the subsequent section, we will provide concise descriptions of each, as they play a vital role in shaping neural networks’ representations.

Supervised Learning

Among the various learning approaches, supervised Learning stands out as the most popular one due to its simplicity and effectiveness, particularly when presented with a moderate to large amount of labeled data. As the name suggests, supervised learning relies on a teacher or a guide to penalize inaccurate outputs from the model. The primary objective of supervised learning is to learn the mapping from an input \( x \) to a corresponding output \( y \). Given a set of \( N \) training samples comprising inputs and outputs \((x_1, y_1), ..., (x_N, y_N)\), the goal is to approximate a function \((f_\theta : X \rightarrow Y)\) that maximizes the similarity between the ground truth value \( y_i \) and the predicted output \( f_\theta(x_i) \) generated by the function. In this process, \( \theta \) represents the set of parameter(s) that collectively influence the function’s outcome. To manipulate these parameters, we employ gradient descent, systematically adjusting \((\theta)\) to minimize a cost function. The cost function
(J) is commonly used to evaluate the performance of the function, with lower values indicating reduced errors on the training data:

\[ J(\theta) = \sum_{i=1}^{m} L(f_{\theta}(x_i), y_i) \]  

(1.11)

Here, \( L \) is a loss function that takes the predicted value \( f_{\theta}(x_i) \) and the corresponding ground truth value \( y_i \) as inputs and quantifies their dissimilarity. Depending on the nature of the data and the specific task, \( X \) and \( Y \) can take various forms. For instance, \( X \) may consist of images, audio, financial data, etc., while \( Y \) could be a categorical variable from a finite set (e.g., \( y_i \in 1, ..., C \) where \( C \) represents the number of classes) for classification tasks, or a real value for regression tasks.

The advantages of this approach, as mentioned earlier, stem from its simplicity. By minimizing the cost function \( (J(\theta)) \) and effectively matching predicted and true values of \( Y \), we can obtain representations that are well-suited for the task at hand. However, this approach is not without its challenges:

- **Performance degradation with increased optimization**: In the process of minimizing the cost function, larger models dealing with complex data, such as images, may learn efficient representations up to a certain point. However, the error may not reach absolute zero. As a result, if the model is trained for an extended period, it might start learning noise from the training data, leading to less efficient representations (i.e., overfitting).

- **Dependency on labeled data**: Since this approach requires labeled data \( (Y) \), its effectiveness is limited by the number of labeled samples available in the dataset. In domains where data annotation is expensive or challenging, this reliance on labeled data poses a generalizability problem that could be catastrophic during deployment.
Unsupervised Learning

Unsupervised learning techniques are learning objectives that discover hidden patterns and structures within unlabeled data. These methods can be pivotal for tasks like representation learning, dimensionality reduction, and learning data distributions. The goal is to infer the natural structure present within a set of data points. Solving the challenge of acquiring meaningful visual representations without relying on supervision has been a persistent endeavor. The majority of established methods can be categorized into two primary classes: generative and discriminative.

Generative Approaches

Auto-Encoders (Figure 1.9) are neural networks with two sub-networks (Hinton & Salakhutdinov, 2006): (i) one that maps a given data sample into lower dimension representation (encoder), and (ii) the other sub-network receives those representations and reconstructs them back into the given data sample (decoder). These family of neural networks are not designed to be generative instead they were aimed to compress high dimensional data into a smaller representation given no labels (i.e. unsupervised representational learning). When network is given a data input $x$, the encoder encodes it into a latent representation $z$ and the decoder uses that latent representation to reconstruct the data input $\bar{x}$. Often to train such network the cost function is defined as the mean squared error between the input $x$ and the reconstructed $\bar{x}$ data sample:

$$f(x, \bar{x}) = \|\bar{x} - x\|^2$$  \hspace{1cm} (1.12)

Although, Auto-Encoders are generally successful at reconstructing data with high quality, often because of the high degree of freedom over the latent code, the training objective leads to a severe over-fitting in the latent space. That is, a small subset of the latent space which is identified by the encoder will yield meaningful content once decoded. However, if a random
1.3. Computational Models of Human Object Recognition

Figure 1.9: Auto-Encoders Architecture. composed of two sub-networks i.e. the encoder and the decoder.

latent code is fed into the decoder, with high probability it will reproduce a meaningless content. Various methods have introduced measures to improve Auto-Encoders.

Another more recent approach that builds upon the concept of AutoEncoders is Masked AutoEncoder (MAE) (He et al., 2021). This approach addresses the limitations associated with traditional Auto-Encoders, particularly their vulnerability to overfitting in the latent space. Developed as a solution to this challenge, MAE enhance the learning process by employing a masking mechanism during training. By strategically masking portions of the input data, this technique encourages the model to capture essential features without becoming excessively focused on details that may lead to overfitting.

**Discriminative Approaches**

Discriminative unsupervised or self-supervised learning objectives exploit priors about the data and treat them as labels. The self-supervised task guides the training through a supervised loss function, however, we do not directly use the loss for final performance evaluation. Rather we investigate the learned intermediate representations at extracting meaningful information from our data. One notable example is contrastive learning.

Contrastive learning primary objective is to acquire an embedding within the input space. This embedding is designed in such a way that the latent representations of two distinct transformations of the same input sample, typically an image, are positioned close to each other. Conversely, when dealing with two transformations of separate input samples, the latent
representations are intended to be distant from each other. These transformations, referred to as data augmentations, encompass a range of operations such as random cropping and resizing to the original dimensions, patch removal, color adjustments, Gaussian noise injection, image blurring, and rotation. Incorporating transformations like color jittering, grayscale conversion, horizontal flipping, and blurring in conjunction with random cropping and resizing has been found to enhance the performance of unsupervised networks in visual recognition tasks (T. Chen, Kornblith, Norouzi, & Hinton, 2020). This approach effectively enhances the network’s capacity to discern meaningful patterns within the visual data.

There are multiple variations of contrastive learning algorithms (X. Chen, Fan, Girshick, & He, 2020; X. Chen, Xie, & He, 2021; M. Caron et al., 2021), one of the most simple and effective methods is SimCLR (T. Chen et al., 2020). SimCLR has been shown to not only outperform previous self-supervised methods on ImageNet. In essence, the framework aims at learning efficient representations by maximizing agreement between differently augmented views of the same data and maximizing difference across contrasting images via contrastive loss in the latent space (Figure 1.10). SimCLR consists of three major modules:

1. **Augmentations Module**: transforms any given data example randomly resulting in two augmented views of the same example, denoted $\hat{x}_i$ and $\hat{x}_j$, which are considered the positive pairs. SimCLR investigated several augmentations for images and their effect on the learned representations. One type of augmentations involves spatial transformations such as cropping and resizing, rotation (Gidaris, Singh, & Komodakis, 2018), flipping, and cutout (DeVries & Taylor, 2017). The other types of augmentations involve appearance transformations, such as color distortion, Gaussian blur, and sobel filtering. Through comprehensive search, (T. Chen et al., 2020) found that random cropping followed by resizing back to the input image size, random color distortion, and random Gaussian blur yield the most efficient representations.

2. **Encoder Module** - $f(\cdot)$: maps the high-dimensional input images to low-dimensional
feature space from augmented examples. Where \( h_i = f(\hat{x}_i) \) and \( h_j = f(\hat{x}_j) \), such that \( h_{i,j} \in \mathbb{R}^d \).

3. **Projection Head Module** - \( g(\cdot) \): maps encoder representations to representations space where contrastive loss is applied. It was shown that MLP with ReLU non-linearity is crucial in this method. Compared to using a linear counterpart or no projection head, the non-linearity significantly improve the quality of the representations obtained. That is, \( z_i = g(h_i) \) and \( z_j = g(h_j) \), such that \( z_{i,j} \in \mathbb{R}^d \). The output dimensionality of \( z \) was shown to not change the performance significantly.

The loss for such objective is termed Normalized Temperature-scaled Cross Entropy Loss (NT-Xent):

\[
l_{i,j} = \frac{\exp \left( \frac{\text{sim}(z_i, z_j)}{\tau} \right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp \left( \frac{\text{sim}(z_i, z_k)}{\tau} \right)}
\]

(1.13)

where \( \mathbb{1}_{[k \neq i]} \in 0, 1 \) is an indicator function returning 1 iff \( k \neq i \), \( \tau \) denotes a temperature parameter (default is 0.5), and \( z \) denotes the encoded representations of a given augmented view. \( N \) is the number of training samples within a mini-batch, \((i, j)\) are positive pairs of each sample. The loss is computed across all positive pairs, in a mini-batch. \( \text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{||\mathbf{u}|| ||\mathbf{v}||} \) denotes the cosine similarity between two vectors \( \mathbf{u} \) and \( \mathbf{v} \).

**Semi-supervised Learning**

Semi-supervised learning encapsulates a category of algorithms that are designed to leverage the benefits of both labeled and unlabeled samples for training. This approach acknowledges the assumption that these samples are often drawn from either the same distribution or closely related distributions. By harnessing the insights gained from both labeled and unlabeled data, semi-supervised learning algorithms aim to enhance their predictive performance and
generalization capabilities. This synergy between labeled and unlabeled data offers a unique opportunity to extract meaningful patterns from the data distribution, even when labeled samples are scarce or costly to acquire. One effective avenue for achieving semi-supervised learning involves building upon the foundation of self-supervised models to create initial representations that encode high-level features within the data. Subsequently, these representations serve as a stepping stone for fine-tuning the model using a limited set of labeled data, culminating in the development of powerful and adaptable semi-supervised learning algorithms. This approach not only addresses the challenges posed by limited labeled data but also capitalizes on the rich source of information embedded in the unlabeled samples.
1.3.5 Invariant Object Recognition

The phenomenon of invariant object recognition, a defining trait of human vision, has captivated researchers from diverse disciplines, driving them to unravel its underlying mechanisms. Early neurobiologically inspired models endeavored to decode the intricate processes occurring within the brain. For instance, early work by C. Cadieu et al. (2007) delved into constructing a model aimed to explain the selectivity and invariance properties observed in visual area V4 of the brain. The model demonstrated that combining feature selectivity and transformation invariance within a hierarchical framework closely emulated the neural responses observed in V4. By replicating the complex responses of V4 neurons to different shapes and transformations, C. Cadieu et al. (2007) found these mechanisms effectively modeled the selectivity of individual V4 neurons to boundary conformation stimuli. Moreover, the model exhibited the same degree of translation invariance observed in V4 and accurately reproduced the observed V4 population responses. Beyond supervised models, C. F. Cadieu and Olshausen (2012) used natural movies to model the underlying structure and statistical regularities of natural visual input, demonstrating the capacity of the visual system to learn intermediate-level representations from unlabelled data.

From its inception, neurobiologically inspired models have paved the way for the development of contemporary neural networks, contributing to the dynamic evolution of computational neuroscience and enhancing our understanding of visual processing mechanisms. While early models (C. Cadieu et al., 2007; C. F. Cadieu & Olshausen, 2012) provided foundational insights into intermediate-level representations, recent advancements have propelled the utilization of DNNs as powerful tools for exploring deeper regions (e.g. IT) within the ventral visual pathway.

Early computer vision work on invariant responses focused on techniques that studied out-of-distribution robustness on image-level distorted or altered in-distribution images (Geirhos et al., 2022, 2020, 2021; Nguyen, Yosinski, & Clune, 2015; Hendrycks & Dietterich, 2019; Kubilius, Kar, Schmidt, & DiCarlo, 2018). These studies aimed to test models’ generalizability...
by altering image statistics relative to the training distribution. While altered in-distribution images provide a measure of resilience that is expected, they do not measure the models’ generalizability to the semantic meaning of the ground truth labels. Furthermore, these image distortions have also been shown to have little to no transfer of robustness to natural distribution shifts (Kubilius et al., 2018). To that goal, numerous generalisation tests have been proposed to directly measure models’ generalizability to out of distribution (OOD) data: ImageNet-R (Taori et al., 2020) which contains various renditions (e.g. art, cartoons) of images for ImageNet classes, similarly, ImageNet-Sketch (Hendrycks et al., 2021) provides sketches, and Stylized-ImageNet (Geirhos et al., 2022) for images with style changes. While measurements of OOD through image-level alterations provide a valuable insight into DNNs’ limitations, most of these techniques unfortunately do not test how these DNNs operate under more naturalistic OOD images.

More recent work investigated naturalistic alterations. Naturalistic alterations are transformations that effect objects in 3D-space and change the global structure of the image. To test the models’ robustness to these naturalistic alterations, previous work proposed naturalistic generalisation datasets: (1) ObjectNet (Barbu et al., 2019) consists of 50,000 real-world images varying object backgrounds, rotations, and viewpoints with 113 classes (total 313 classes) overlapping with ImageNet (Deng et al., 2009), (2) ImageNet-Vid (Shankar et al., 2019) consists of 57,897 images and exploits the temporal component of videos to derive naturalistic alterations with 293 classes, and (3) SI-SCORE (Djolonga et al., 2021) consists of 611,608 synthetic images varying object size, location, and rotation in 2D-space with 614 foreground objects from 62 classes. While these datasets provide an extensive measurement for robustness on naturalistic alterations, they either lack in number of alterations (e.g. light, material changes), number of factor of variations (e.g. degrees of object rotation), or objective control on factor of variations (i.e. sampling uniformly across different degrees of object rotation).

Other research efforts have focused on exploring the robustness of DNNs using 3D graphics renderers (Alcorn et al., 2019, Abbas & Deny, 2022, Madan et al., 2021, Ruiz, ...
In contrast to datasets with naturalistic alterations, these approaches involve transformations that impact the visual characteristics of objects while keeping the semantic features unchanged. This distinction becomes particularly challenging to achieve at a large scale when working with image datasets. Although earlier studies have indicated that DNNs lack robustness to naturalistic 3D transformations, these shortcomings have primarily been observed in relation to changes in object orientation (Alcorn et al., 2019; Abbas & Deny, 2022; Madan et al., 2021; Ruiz et al., 2022). A recent study by Ibrahim, Garrido, Morcos, and Bouchacourt (2022) demonstrated that SOTA (State-of-the-art) models struggle with common alterations in object orientation, size, and background. Moreover, these investigations do not encompass the examination of how human invariant object recognition responds to such naturalistic alterations.

### 1.4 Thesis Objectives

In this thesis, we investigate object recognition performance of deep neural network models and humans under 3D naturalistic transformations. While this question has been extensively examined in psychology and neuroscience (humans) (Biederman, 1987; Bülthoff et al., 1995; DiCarlo et al., 2012), and more recently, in computer vision (models) (Abbas & Deny, 2022; Ruiz et al., 2022; Ibrahim et al., 2022), integration of results across fields is lacking. One major obstacle is the lack of a stimulus set that enables fair comparison between humans and models on object recognition under 3D naturalistic transformations (Geirhos et al., 2020). Another is a discrepancy of goals between the two fields (Hassabis et al., 2017). Computer vision research focuses on improving model performance on specialized visual tasks, while human research aims to understand how the visual system supports object recognition across a range of tasks.

This thesis has the overarching aim to investigate invariant object recognition at the intersection of psychology, neuroscience, and computer vision. In doing so, we anticipate that outcomes will mutually benefit these fields, and help push towards a broader adoption of an
integrated empirical-computational framework for investigating human vision. Within such a framework, we can use DNNs as a modeling platform to better understand which type of network organization (e.g. hierarchical system, convolutional filters) and which developmental factors (e.g. visual diet, learning objectives) are most relevant for simulating human behaviour on visual tasks. Insights acquired in this endeavor can in turn be used to improve DNNs as models of the human visual system, and with that, may also create more versatile DNNs that perform well across a range of natural visual tasks, which is relevant for applications such as self-driving cars.

We address our overarching aim by investigating object recognition performance of humans and DNNs on the same stimulus set. For models, we investigate multiple transformations: object rotation, object scaling, and background change. For humans, we focus on object rotation in depth, which is (one of) the most challenging transformations to master (Gliga & Dehaene-Lambertz, 2007; Nishimura et al., 2015). We studied the effects of the transformations in-depth: (1) To understand which naturalistic variations models or humans are not robust to, we measure object recognition performance across variations, and compare models and humans on accuracy and consistency of errors, (2) To understand what makes some models more robust to these variations than others, we perform an in-depth investigation into which components of these models (i.e. model architecture, visual diet, and learning objective) drive model performance.

In summary, the objectives we achieved in this thesis are:

1. We built an image generation pipeline that, given 3D objects, can generate high-quality synthetic test datasets for all three naturalistic 3D transformations.

2. We developed a stimulus set that systematically introduces a range of 3D object transformations (including object rotation in depth) and that enables direct comparison between humans and DNNs.

3. We measured and compared object recognition performance on our stimulus set between
humans and DNNs, and found that (1) DNNs are less invariant to object rotation in depth than humans are, (2) DNNs and humans make different errors, and (3) DNNs generally do not perform well on invariant object recognition, although some transformations appear more challenging than others.

4. We examined which design features (i.e. model architecture, visual diet, and learning objective) contribute most to performance differences between DNNs.
Chapter 2

Methods

2.1 Stimuli

2.1.1 Category Mapping for Fair Comparison

DNNs and humans exhibit distinct tendencies when recognizing and categorizing objects, possibly influenced by their training datasets and innate cognitive mechanisms. Specifically, DNNs, when trained on comprehensive datasets like ImageNet (Deng et al., 2009), delve into an array of detailed and fine-grained categories. Unlike DNNs, human behavioural tendencies generally lean towards broader, or what might be termed "entry-level," categorizations. For a genuine and insightful comparison of object recognition capabilities between DNNs and humans, it becomes imperative to address and reconcile this categorization divergence. Such a reconciliation ensures that both entities are assessed on equivalent grounds, thereby facilitating a more accurate comparison of their object identification.

To achieve this alignment in categorization, we drew inspiration from a mapping approach proposed by Geirhos et al. (2021, 2018). While the ImageNet dataset, encompassing
1,000 nuanced categories, may classify a canine entity as a specific breed like "German shepherd", human intuition typically resonates with a more generalized label such as a "dog" (Rosch et al., 1976). Guided by this observation, Geirhos et al. (2018) transformed the nuanced ImageNet categories into broader classifications, leveraging the semantic capabilities of the WordNet hierarchy (Fellbaum, 1998). The "16-class-ImageNet" dataset, elegantly distills 1,000 specific ImageNet categories into 16 overarching, or "entry-level" categories. The categories encompass: airplane, bicycle, boat, car, chair, dog, keyboard, oven, bear, bird, bottle, cat, clock, elephant, knife, and truck.

### 2.1.2 3D Object Models

The primary aim of sourcing 3D object models was to obtain a rich collection that closely represents real-world objects, catering specifically to the 16 generalized or "entry-level" categories (Figure 2.1). The key criteria for object selection included: **Relevance to the 16 Classes**: Only object models that fit within the parameters of the 16 overarching categories (airplane, bicycle, boat, car, etc...) were chosen. This ensured a direct correlation with the previously discussed categorization harmonization. **Realistic Quality**: Each object model had to be of high resolution and detail. The objective was to source models that, when transformed and visualized, would closely mimic real-world objects in their intricacies and nuances.

![Figure 2.1](image.png)

**Figure 2.1**: An overview of how 3D object models were sourced, pre-processed, and used to generate images for evaluating humans and computational models.
Chapter 2.

Figure 2.2: Overview of the selected 3D object models. The objects are ordered, such that columns reflect categories and rows reflect exemplars. These objects were used in experimental trials.

Figure 2.3: Overview of the 3D object models that were used in practice trials.

We sourced 176 3D object models from prominent online repositories of 3D objects: sketchfab, turbosquid, blenderkit, and cgtrader. We sourced 11 3D objects for each of the 16 classes, 10 for experimental blocks (Figure 2.2) and 1 for practice blocks (Figure 2.3). Furthermore, to ensure uniformity and standardization across the 3D objects for consistent evaluation, we performed the following preprocessing on each 3D object:

1. **Uniform Scaling**: is employed to ensure consistency in the size of 3D objects across the dataset. This guarantees that size variances don’t introduce biases or discrepancies in object recognition, allowing for a fair evaluation across 3D objects. For every individual
2.1. Stimuli

A 3D object model, a bounding box is established. This box encapsulates the entirety of the object along the x, y, and z axes, essentially serving as a three-dimensional frame around the object. The bounding box outlines the outermost of the object in each of the three dimensions. Out of the three axes, the largest dimension is identified and used as the primary reference point for the scaling process. If the object’s maximum dimension is larger than the standardized size, the object is scaled down proportionally. Conversely, if it’s smaller, the object is scaled up. The pivotal aspect here is proportionality. All three dimensions are scaled at the same rate to maintain the original proportions and integrity of the 3D object model. This ensures that while the overall size of the object changes, its relative proportions remain intact, preserving the object’s authenticity.

2. Object Centering: aims to standardize the position of 3D models within a given frame. This step ensures that each object is optimally positioned for rendering. A translation operation is performed on the 3D object to align with the center of the camera.

2.1.3 3D Transformations & Stimulus Generation

We utilized ThreeDWorld (TDW) (Gan et al., 2021), a 3D simulation platform based on the Unity game engine to generate synthetic image datasets. Figure 2.4 shows the TDW space, including the locations of the object and camera. The following naturalistic 3D transformations were applied to the 3D object models and environment to create our synthetic stimuli (Figure 2.5):

- **Object Rotation**: objects are placed with their centers at the world origin and rotated a full rotation around each world axis, one axis at a time, and images are captured after each angle change (step size = 45°) from a camera distanced 1 unit distance away with an elevation of 0°from the xz plane and and azimuth of 0°from yz plane. For each axis of rotation, this procedure yields 8 views, including the reference view.

- **Object Scale**: controls the size of the 3D object by varying the distance of the camera to
Figure 2.4: Schematic of the TDW space, showing object and camera location, as well as images generated by the camera before object rotation (reference) and after object rotation about each axis.

the object, with the one factor of variation being camera distance. The camera by default is set to 1 unit distance away from the camera, setting camera distance < 1 results in a zooming effect. When too close, part of the object can be out of camera view. We move the camera in the range of [0.3, 3] unit distance, covering close and far snapshots. As a consequence, object size varies between 200 and 12.5 percent of the size captured by the reference image. Size here corresponds to the maximum of the object’s height or width in the image. We examine 8 object sizes, including the reference size.

- **Background**: changes the scene’s background by changing its high-dynamic-range image (HDRI). We use a set of 7 HDRIs displaying a variety of indoor and outdoor scenes. Following (Abbas & Deny, 2022), we use a grey default background with RGB pixel values of (0.485, 0.456, 0.406), corresponding to the average pixel color of ImageNet images. The default background is used for all augmentations with the exception of the
2.1. **Stimuli**

background change augmentation.

Canonical/reference views of 3D objects are used to compute baseline performance (Figure 2.5). The default value for each 3D transformation is applied during canonical view generation. Each 3D transformation can have one or more factors of variations. For each factor of variation, we generated 8 synthetic images. In total for each factor of variation, there are 16 categories x 10 exemplars x 8 images = 1,280 generated images. Across all five factors of variation, this results in 6,400 generated images. We saved the generated images in .png format at a resolution of 224 x 224 pixels. We additionally created 1/f noise masks (see Figure 2.6 for an example) using Kendrick’s Kay publicly available MATLAB code base (https://github.com/cvnlab/knkutils).

Figure 2.5: An overview of all transformations used to generate images. The black border highlights object rotation in depth about the x axis, the transformation that we used for the human experiment.
2.2 Experimental Design

2.2.1 Participants

Every participant possessed either standard vision or vision corrected to normal standards and reported no previous neurological disorders. The research received approval from Western University’s HSREB. Before engaging in the study, all individuals provided their implied consent via a questionnaire on Qualtrics (Qualtrics, Provo, UT). Participants were paid 15$ per hour for participating in the experiment. A total of 17 participants were recruited for the study. Two participants were excluded from the study as their performance fell below 80% on the second practice block. The remaining 15 participants consist of 10 male and 5 female individuals with age ranging from 22 to 28 years and a mean age of 25 years.

2.2.2 Human Experiment

We constructed the experimental protocol using the PsychoPy software (Peirce et al., 2019). We recruited participants from the Western University community. We sought individuals aged between 18 and 35, with either standard vision or corrected-to-normal eyesight, and no prior incidents of neurological disorders. After recruitment, participants were channeled to Qualtrics for implied consent, followed by a questionnaire, which asked participants to indicate any history of neurological disorders. Subsequently, they were redirected to the experiment which was made available on Pavlovia (Open Science Tools, Nottingham, UK).

On Pavlovia, we used a credit card scaling procedure and blind spot estimation to estimate the screen resolution and the participant’s distance to the screen, respectively (Brascamp, 2021). Subsequently, participants were given explicit instructions about the task and on how to respond using a mouse click. They were encouraged to answer as accurately as possible and to
rely on their best judgment when in doubt. To ensure a complete understanding of the task, they were advised to acquaint themselves with all 16 categories presented on the response screen.

To begin, the first and second practice blocks utilized a random selection of 256 ImageNet validation images. The ImageNet validation images are photos of real-world objects belonging to one of the 16 categories. This approach served two primary purposes: firstly, to familiarize participants with the nature of the task, and secondly, to filter out any participant who wasn’t paying adequate attention. While the first practice block offered visual feedback by highlighting the correct category and displaying either a red cross for incorrect/missed answers or a green check mark for correct answers, the second block abstained from providing any feedback. If a participant’s accuracy fell below 80% on the second practice block, they were excluded from the study. The third practice block consisted of images generated using the ThreeDWorld platform (Figure 2.3 shows the 3D objects used to generate the practice images).

For the first two practice blocks, we employed Python for image preprocessing. Within the pool of ImageNet images representing the 16 entry-level categories, we eliminated grayscale images (constituting 1%) as well as images that were smaller than $256 \times 256$ pixels and images that showed multiple objects from the 16 categories. Our approach involved cropping all images to a central patch with dimensions of $224 \times 224$ pixels, following a two-step process. Initially, each image was cropped to the largest center square feasible. Subsequently, this center square was downsized to the intended dimensions using the PIL.Image.thumbnail((224, 224)) method.

The main experiment, comprising 10 blocks (128 trials per block), diverged from the first two practice blocks in its image source. The images for these blocks were generated using the ThreeDWorld platform, by applying object rotation in depth on our stimulus set of 3D objects (see Figure 2.2 and Figure 2.5). In a bid to maintain the participants’ enthusiasm and commitment to the task, a summary of their performance was displayed on the screen at the end of each block. After each block participants were given a mandatory 30-seconds break after which they were offered the option to continue after a mouse click. In total, the experiment took
approximately 1.5 hours to complete (Figure 2.6).

On each trial, a grey screen with a centralized white fixation cross was first presented. Participants were instructed to initiate stimulus onset by clicking on the fixation cross. Subsequent to this action, an image was displayed for a duration of 200 ms. The image was immediately followed by a colored mask for 200 ms, which served to increase task difficulty and reduce effects of recurrent processing on performance. The latter enable a fairer comparison between human participants and DNNs, whose architectures only allow for feedforward flow of information (Geirhos et al., 2020). The next phase involved presenting the participants with a response screen with multiple category buttons. This remained available for a maximum period of 1500 ms or until a selection was made by the participant, whichever was shorter. The category buttons were arranged in a circular configuration, ensuring equal distance from the central fixation cross to preclude any spatial biases in the decision-making process. Furthermore, the position of the mouse was locked during the presentation of the stimulus and mask. During the response phase, participants were tasked with selecting the category they deemed most congruent with the stimulus (Figure 2.6).

2.2.3 Neural Network Experiments

We tested object recognition performance under 3D naturalistic transformations for a set of 65 feedforward DNNs. All DNNs were pretrained on ImageNet-1K (Deng et al., 2009) or ImageNet 21-K (Ridnik, Ben-Baruch, Noy, & Zelnik-Manor, 2021), which consist of 1.28M and 14M images, respectively. We selected our set of DNNs such that they captured variability in the design features of interest: architectural base, model complexity, visual diet, and learning objective (Figure 2.7). Our aim is to determine the relative contribution of these design features in explaining DNN performance on visual tasks that probe invariant object recognition. Architectural base can be a convolutional network (Krizhevsky et al., 2012; He, Zhang, Ren, & Sun, 2016a), a vision transformer network (Dosovitskiy et al., 2021), or a combination of the
2.2. **Experimental Design**

**Figure 2.6**: An overview of the human experiment. Participants were instructed to categorize object images in a forced-choice paradigm. (a) Each trial started with a fixation cross that participants to click on, followed by image and mask presentation for 200 ms each, and a response screen that was presented for 1500 ms or until the participant clicked on one of the category icons, whichever came earlier. Images and masks were presented at 5 degrees of visual angle. (b) Schematic of the experiment, which consisted of three practice blocks and 10 experimental task blocks. Unknown to the participants, the second practice block served as a test: only participants with an accuracy of 80 percent or higher were included in the experiment and invited to continue.

Model complexity refers to the number of model parameters and can be small, base, or large. Visual diet refers to the number of training images, and is either 1.28M or 14M images. Learning objective refers to the loss function used during training, and can be supervised (Krizhevsky et al., 2012), semi-supervised (Bao, Dong, Piao, & Wei, 2022), or self-supervised (He, Fan, Wu, Xie, & Girshick, 2020). We also ensured that the selected DNNs included vanilla networks commonly used in computational neuroscience, such as AlexNet (Krizhevsky et al., 2012), and SOTA networks, commonly used in computer vision, such as Bidirectional Encoder representation from Image Transformers (BEiT) (Bao et al., 2022).

To test model performance, each test image was first resized to match the input size of the neural network model. This resizing step ensures that the image dimensions are consistent.
with the model’s expected input size. The resizing process maintains the image’s aspect ratio while fitting it into the designated input dimensions. The pixel values of the resized image were normalized before being fed into the model. Normalization involves scaling the pixel values to have zero mean and unit variance. This step ensures that the input data falls within a certain range and is centered around zero, which facilitates stable and efficient training and inference in neural networks.

Once the resized and normalized image was prepared, it was inputted into the neural network model. The model’s forward pass was then performed, which involves passing the input image through the layers of the network to generate a prediction or output. This output is often a probability distribution over the different categories that the model has been trained to recognize. During the evaluation process, the weights of the neural network model were frozen. This means that the model’s parameters remained fixed and were not updated based on the input image. Weight freezing is essential to maintain consistency between the training and evaluation phases. It ensures that the model’s performance on the test images is reflective of its generalization capabilities rather than its ability to fine-tune to the specific data.

The model’s prediction for the test image was compared to the ground truth label of that image. This comparison allowed the calculation of accuracy, which measures the percentage of correctly predicted categories out of the total tested images. This accuracy score served as a quantitative measure of the model’s performance under the specific transformation being evaluated.

### 2.2.4 Error Consistency

Error consistency measures how closely machine decisions align with human decision-making behavior [Geirhos et al., 2021]. Beyond just average accuracy, error consistency evaluates the degree of agreement between human and machine errors. It’s crucial to determine if the alignment is greater than what might occur by chance. For example, if two decision-
2.2. Experimental Design

Figure 2.7: An overview of all design features that DNNs were varied across: architecture, learning objective, and visual diet. Architectural changes can be varied either through architectural base (convolutional vs vision transformer) or model complexity (small, base, large).

makers each have a 95% accuracy rate, their decisions will appear to be at least 90% consistent, even if their respective 5% errors do not overlap. This high observed consistency can occur because both parties are correct on the majority of the test data. Error consistency, also known as Cohen’s kappa [Cohen, 1960], denoted $\hat{o}_{h,m}$, measures whether the observed consistency exceeds that which would be expected from two independent decision-makers with equivalent accuracy. The average error consistency is then determined by

$$E(m) : \mathbb{R} \rightarrow [-1, 1], m \mapsto \frac{1}{|D|} \sum_{d \in D} \frac{1}{|H_d|} \sum_{h \in H_d} \frac{1}{|C_d|} \sum_{c \in C_d} \left( \frac{1}{|S_{d,c}|} \sum_{s \in S_{d,c}} b_{h,m}(s) - \hat{o}_{h,m}(S_{d,c}) \right) \frac{1}{1 - \hat{o}_{h,m}(S_{d,c})}$$ (2.1)
where \( s \in S_{d,c} \) denotes a sample \( s \) of condition \( c \) from dataset \( d \), \( b_{h,m}(s) \) is the observed consistency which has one if both a human observer \( h \) and a machine observer \( m \) decide either correctly or incorrectly on a given sample \( s \), and zero otherwise. Equation (2.1) computes the average error consistency across datasets, participants/models, and conditions in that order, respectively.
Chapter 3

Results

3.1 Invariant Object Recognition in Humans and Computational Models

Figure 3.1 presents a comprehensive overview of the performance outcomes achieved by over 60 neural network models, with human participants’ performance indicated in red, in the context of object rotation in depth (see example stimuli in Figure 2.5). In the figure, the background grey bars serve as a representation of the baseline performance of models for canonical views. On the other hand, the dark foreground bars depict the average performance encompassing various transformations. For models, the standard error of the mean has been calculated across all images, while for human participants, it has been computed across participants. The visualization in Figure 3.1 unveils an interesting observation: unlike computational models, human participants demonstrate not only accurate performance on canonical views but also exhibit a minimal drop in performance when confronted with object rotation along the x-axis. This stands in contrast to computational models, which, despite being able to match or sometimes even surpass human performance on canonical views, face challenges in maintaining accuracy when faced with object rotation along the x-axis. This discrepancy highlights an intriguing distinction between
Figure 3.1: An overview of the performance of more than 60 models and human participants concerning the 3D transformation involving object rotation along the x-axis. The background bars in grey illustrate the baseline performance of models when presented with canonical views. The dark foreground bars in the foreground represent the average performance across various transformations. The text placed below the bars provides the names of the corresponding models. The red bar highlights the average performance achieved by human participants. The dashed line signifies change level accuracy.

The performance of computational models and the adaptability of human visual recognition in handling object transformations.

3.2 Computational Models Limitations vs Humans

Figure 3.2 illustrates the collective performance of more than 60 computational models across various naturalistic 3D transformations, including object rotation along the x, y, and z axes, background change, and object scale adjustments. Computational models in Figure 3.2 are shown to struggle with naturalistic 3D transformations such as object rotations, especially along...
the x and z axes. This is due to rotations along the y-axis often result in transformations that are more likely to be similar in visual characteristics to canonical views than those along the x and z axes (depth planes).

Figure 3.3 offers a detailed visualization of the distribution of models and human performance on 16 categories, showing the average performance gap (i.e. performance for canonical - transformed object views) for models (x axis) and humans (y axis). Figure 3.3 was computed to demonstrate if humans and models find the same categories challenging. We found that models and humans find difficulty with different categories. For instance, models on average perform worse on the Truck and Elephant categories, while, humans perform worse on the Clock and Oven classes.

The disparity in performance across different object categories can be attributed to the varying degrees of complexity inherent to each category’s visual features. Certain categories may possess visual characteristics that are challenging for computational models to accurately recognize under specific 3D transformations. For example, the "boat" category might involve intricate shapes or variations that make it difficult for models to maintain accurate recognition across diverse rotations, scales, or backgrounds. On the other hand, categories like "bird" could possess more distinct visual attributes that allow models to maintain better recognition performance under similar transformations.

Furthermore, the inherent complexity of object categories can impact the models’ ability to generalize learned features effectively. Models might excel at categories with clear and distinctive features that are consistent across different perspectives and scales, contributing to better performance. In contrast, categories with more complex and diverse visual characteristics may challenge the models’ capacity to learn invariant representations, leading to lower accuracy.

In a similar vein, Figure 3.4 depicts the performance of human participants when confronted with the 3D transformation involving object rotation along the x-axis. The bars within the figure delineate performance across (a) all categories, (b) the category with the lowest
Figure 3.2: Average performance across all models on all 3D transformation. The dashed line represents the accuracy achieved on the canonical view, offering a baseline reference for comparison.

Figure 3.3: Average performance gap (i.e. performance for canonical - transformed object views) for models (x axis) and humans (y axis) across 16 categories.
3.3. Effect of Architectural Base and Model Complexity on Performance

performance, and (c) the category with the highest performance. The dashed line functions as a benchmark, signifying the accuracy achieved on the reference view for all categories, thereby facilitating a comparative assessment. Strikingly, as demonstrated in both Figure 3.4a and Figure 3.4b, individuals face challenges with the "oven" category while excelling in the "knife" category during object rotation along the x-axis. This variability in performance underscores the intricate nature of human object recognition, influenced by the specific visual attributes and cognitive processes associated with each category. Furthermore, this performance discrepancy is also mirrored in participants’ reaction times; better performance on certain categories is associated with slower reaction times, reflecting the dynamic interplay between accuracy and speed of recognition (Figure 3.4a).

The dissimilarities in performance between computational models and humans across various categories can be attributed to the dissimilarity in how the two entities process visual information. The innate perceptual abilities of humans enable them to excel at certain categories, while computational models might struggle due to the absence of certain human-like cognitive mechanisms. Factors such as contextual understanding, feature extraction, attention mechanisms contribute to variations in performance. As a result, the strengths and limitations of computational models and humans can manifest differently across different object categories, resulting in varying levels of performance on tasks like object rotation along the x-axis.

3.3 Effect of Architectural Base and Model Complexity on Performance

Figure 3.5 demonstrates the relationship between architectural base and model complexity in influencing object recognition performance during object rotation along the x-axis. Figure 3.5 showcases three different architectures: Convolutional Networks (Conv), Vision Transformers (ViT), and a hybrid architecture combining Convolutional and Vision Transformer elements
Figure 3.4: A summary of human participants’ performance (a) and reaction time (b) when encountering the 3D transformation involving object rotation along the x-axis. The bars represent performance/reaction time across (1) all categories, (2) the category with the lowest performance/reaction time, and (3) the category with the highest performance/reaction time. The dashed line signifies accuracy/reaction time achieved on the reference view for all categories, providing a benchmark for comparison.

(ConViT). Each architecture spans three levels of model complexity, categorized as Small (S), Big (B), and Large (L). The background grey bars indicate the baseline performance of models on canonical views, while the foreground color bars represent the average performance across all object rotations along the x-axis, with different colors corresponding to specific model categories.

Overall, the results indicate that larger models tend to exhibit better performance compared to their smaller counterparts when subjected to object rotation along the x-axis. This trend becomes more pronounced as model complexity increases. While model architecture also plays a role in performance, the effect is relatively smaller compared to model complexity. Among the architectural categories, Deit3 (ViT) shows the highest performance, followed by ConvNext (Conv), and ConViT. While models generally exhibit similar performance on reference views, the introduction of object rotation along the x-axis accentuates the impact of model complexity and architecture on recognition accuracy, demonstrating the intricate interplay between these factors in shaping model performance under 3D transformations.
3.3. Effect of Architectural Base and Model Complexity on Performance

Figure 3.5: Summary of 8 models’ and humans’ performance on object rotation along the x-axis 3D transformations, highlighting the interplay between architecture and model complexity. Three architectures, namely Conv, ViT, and ConViT, are presented, each with three levels of model complexity: Small (S), Big (B), and Large (L). The background grey bars denote baseline performance on canonical views, while foreground colored bars depict average performance across all object rotations on the x-axis 3D transformations. Different colors correspond to specific model architectures, except the red bar which displays average participants’ performance.
In Figure 3.6, we investigate deeper into which type of images yield the drop in performance. Figure 3.6 presents a comprehensive analysis of the performance of 8 models across four distinct 3D transformations—object scale, object rotation along the x, y, and z axes. This analysis aims to shed light on the interaction between model complexity, architecture, and the capacity to handle varied degrees of 3D variations. We consider three model architectures—Convolutional Networks (Conv), Vision Transformers (ViT), and a hybrid ConViT architecture—each represented at three complexity levels: Small (S), Big (B), and Large (L). The dashed bottom line in the sub-figures signifies the performance at the chance level, while the two dashed lines at the top reflect the range of performance observed on reference views.

The observed trends provide insights into how models handle different 3D transformations. On the matter of object scale transformations, it becomes evident that models struggle more when presented with zoomed-out images (12.5% zoom-out) compared to highly zoomed-in images (200% zoom-in). The fact that models exhibit lower performance when drastically zooming out, even though the shape of the object is near non-existent at 200% zoom-in, suggests that models may be demonstrating a texture bias rather than accurately capturing the shape.

In the context of object rotation along the x-axis, both models and human participants demonstrate a parallel pattern of performance as the degree of rotation varies, humans consistently surpass the model’s accuracy, as observed in Figure 3.5. Furthermore, both models and humans face challenges when objects are rotated to their lateral sides, leading to a notable drop in performance. This challenge becomes even more pronounced when objects are observed from unconventional angles, such as the top or bottom. For instance, consider a car observed horizontally, revealing only its top or bottom parts—both models and humans exhibit reduced performance in such cases. This demonstrate the significance of contextual cues within the scene for effectively processing these distinctive viewpoints. The trends persist in object rotation along the y-axis, with models following a similar trajectory. Two significant performance dips occur in this scenario, possibly linked to object categories like clocks or ovens, where models may lack exposure to the back of these objects during training, introducing a bias that affects
3.3. Effect of Architectural Base and Model Complexity on Performance

Figure 3.6: Summary of 8 models’ performance on four 3D transformations (object scale, object rotation on x, y, and z axes) across 8 degrees of variation, highlighting the interplay between model complexity and architecture. Three architectures, namely Conv, ViT, and ConViT, are presented, each with three levels of model complexity: Small (S), Big (B), and Large (L). The dashed line at the bottom denotes the performance at chance level, while the two dashed lines at the top of each sub-figure represent the range of performance observed on reference views.

recognition accuracy. Finally, object rotation along the z-axis, an axis that doesn’t alter the perspective of the object but affects its upside position, reveals a lack of significant dips in performance. While models maintain stable performance, a slight decrease can be observed, likely attributed to the models’ limited exposure to objects in upside-down orientations during training.
3.4  Effect of Model Visual Diet on Performance

Figure 3.7 elucidates the intricate interplay between model complexity and visual diet in shaping the efficacy of object recognition during object rotation along the x-axis. Figure 3.7 showcases Vision Transformers (ViT) architecture, spanning three tiers of model complexity categorized as Small (S), Big (B), and Large (L), each of which is further trained with differing amounts of data—1.28 million images versus 14 million images. The background grey bars indicate the baseline performance of models on canonical views, while the foreground colored bars represent the average performance across all object rotations along the x-axis 3D transformations, with different colors corresponding to specific model complexities. Darker colors indicates models that are trained on more data i.e. 14 million vs 1.28 million images.

The main result is that larger models generally perform better than smaller ones when dealing with object rotation along the x-axis. This effect becomes stronger when models are trained with a larger dataset of 14 million images. This finding aligns with the observations in Figure 3.6 which also shows that model complexity influences object recognition performance.

Figure 3.8 presents a comprehensive analysis of the performance of 6 models across four distinct 3D transformations—object scale, object rotation along the x, y, and z axes. This analysis aims to shed light on the interaction between model complexity, visual diet, and the capacity to handle varied degrees of 3D variations. We consider one model Vision Transformers (ViT), represented at three complexity levels: Small (S), Big (B), and Large (L), either trained with 1.28 million vs 14 million images. The dashed bottom line in the sub-figures signifies the performance at the chance level, while the two dashed lines at the top reflect the range of performance observed on reference views.
Figure 3.7: Summary of 6 models’ and humans’ performance on object rotation along the x-axis in 3D transformations, highlighting the effect of visual diet on performance. One architecture, namely ViT, is presented, with three levels of model complexity: Small (S), Big (B), and Large (L) and either trained with 1.28 million vs 14 million images. The background grey bars denote baseline performance on canonical views, while foreground colored bars depict average performance across all object rotations on the x-axis 3D transformation. Different colors correspond to specific model architectures, except the red bar which displays average participants’ performance.
Figure 3.8: Summary of 6 models’ performance on four 3D transformations (object scale, object rotation on x, y, and z axes) across 8 degrees of variations, highlighting the interplay between model complexity and architecture. One architecture, namely ViT is presented, each corresponding with three levels of model complexity: Small (S), Big (B), and Large (L) and either trained with 1.28 million vs 14 million images. The dashed line at the bottom denotes the performance at chance level, while the two dashed lines at the top of each sub-figure represent the range of performance observed on reference views.
3.5 Effect of Learning Objective on Performance

In the context of investigating the effect of learning objectives on performance, the results presented in Figure 3.9 offer insights into the performance of 6 models and humans in object rotation along the x-axis within 3D transformations. The analysis centers on one architectural type, Vision Transformers (ViT), encompassing three levels of model complexity: Small (S), Big (B), and Large (L), all trained with varying learning objectives—supervised, semi-supervised, and self-supervised. The background grey bars provide a baseline measure of performance on canonical views, while the foreground colored bars signify the average performance across all object rotations on the x-axis 3D transformations. Each color corresponds to specific model architectures, excluding the red bar which represents the average performance of participants.

A notable trend is observed among the general results, emphasizing the impact of learning objectives on performance. Semi-supervised models exhibit the most favorable performance, notably approaching human performance levels in object rotation along the x-axis. This trend suggests that the utilization of semi-supervised learning objectives enhances the models’ ability to capture intricate patterns and features necessary for recognizing objects in transformed scenarios.

The results showcased in Figure 3.10 provide a comprehensive overview of the performance exhibited by 6 models across four distinct 3D transformations, namely object scale and rotation along the x, y, and z axes. The focus of this analysis is to illuminate the intricate interplay between model complexity and learning objectives. The architectural framework under investigation is the Vision Transformer (ViT), featuring three levels of model complexity: Small (S), Big (B), and Large (L), each trained with supervised, semi-supervised, and self-supervised learning objectives. The dashed lines at the lower boundary indicate chance-level performance, while the upper boundary features two dashed lines representing the range of performance observed for reference views.
Figure 3.9: Summary of 6 models’ and humans’ performance on object rotation along the x-axis in 3D transformations, highlighting the effect of learning objective on performance. One architecture, namely ViT, is presented, with three levels of model complexity: Small (S), Big (B), and Large (L) and trained with supervised, semi-supervised, self-supervised learning objective. The background grey bars denote baseline performance on canonical views, while foreground colored bars depict average performance across all object rotation on the x-axis 3D transformation. Different colors correspond to specific model architectures, except the red bar which displays average participants’ performance.
3.6 Error Consistency between DNNs and Humans

Object rotation along the x-axis reveals intriguing trends. Notably, the performance of semi-supervised models surpasses that of human participants for object rotation on x-axis in scenarios close to the reference view. However, this superiority diminishes significantly for other variations. Additionally, in terms of object rotation on the y-axis and object scale, semi-supervised models achieve parity with supervised models. Remarkably, on object rotation along the x and z axes, the semi-supervised models outperform supervised models. This aligns with the growing understanding that semi-supervised learning objectives can lead to the extraction of more robust and versatile features that contribute to better performance across various transformation scenarios.

3.6 Error Consistency between DNNs and Humans

Figure 3.11 incorporates error consistency analysis proposed by Geirhos et al. (2021), which evaluates whether behavioural responses from two given entities exhibit the same errors. Here, we performed a comprehensive comparison among all models and human participants. By evaluating whether models and humans commit errors on the same images, we gain insights into the degree of congruence in their error patterns. Interestingly, our findings reveal a distinct clustering of models, irrespective of their learning objective, visual diet, or model complexity. Comparing models with each other reveals systematic errors in a highly consistent manner, with some exceptions being pixel-level trained models (such as MAE), and models that are very low on model complexity (such as alexnet). Conversely, the error patterns exhibited by humans significantly differ from all models. The error inconsistency between human participants and models can be further seen in their respected average confusion matrix Figure 3.12.
Figure 3.10: Summary of 6 models’ performance on four 3D transformations (object scale, object rotation on x, y, and z axes) across 8 degrees of variations, highlighting the interplay between model complexity and learning objective. One architecture, namely ViT is presented, each corresponding with three levels of model complexity: Small (S), Big (B), and Large (L) and trained with supervised, semi-supervised, self-supervised learning objective. The dashed line at the bottom denotes the performance at chance level, while the two dashed lines at the top of each sub-figure represent the range of performance observed across all human participants on reference views.
3.6. Error Consistency between DNNs and Humans

Figure 3.11: Analysis of error consistency for in-depth object rotation, revealing a gradient from low consistent errors (depicted in lighter colors) between human participants and models, to moderately consistent errors among human participants, to highly consistent errors (depicted in red) among DNN models. The similarity in error consistency patterns among DNNs illustrates the shared challenges they face in recognizing objects under varying degrees of in-depth rotation.
Figure 3.12: Averaged confusion matrices across models (left) and human participants (right) for 17 categories, the last category being 'none'. 'None' only used in human experiments for cases where participants did not provide a response. Colorbar indicates the number of conditions (80 conditions per category).
3.7 Explained Variance in Model Performance through Regression Analysis

In the context of assessing the factors influencing model performance, Figure 3.13 presents a comprehensive analysis of the explained variance achieved through linear regression. We employed learning objectives, visual diet, model complexity, and model architecture as predictor variables to ascertain their contributions to the observed variability in model performance across all 3D transformations.

When assessing the contribution of model architecture in isolation, the results reveal a marginal effect, yielding an explained variance near zero. This outcome suggests that the intrinsic architectural distinctions between models may not be the primary driver of the observed variation in their performance. In contrast, model complexity emerges as a moderate determinant, accounting for approximately 20% of the variability in model performance. This indicates that as the complexity of models increases, there is a corresponding but partial elevation in their performance.

The influence of learning objectives surfaces as a more influential factor, contributing to roughly 50% of the variability in model performance. The specific learning approach adopted plays a pivotal role in shaping the models’ capacity to generalize effectively across 3D transformations. Furthermore, the breadth and diversity of the training data, represented by the concept of visual diet, wield a substantial influence on the explained variance, elucidating about 35% of the variability in model performance. A more comprehensive visual diet equips models with enhanced adaptability and performance across varying scenarios.
Figure 3.13: Analysis of explained variance in model performance achieved through linear regression, utilizing model architecture (base and complexity), visual diet, and learning objective as predictor variables. Architectural base reflects whether a DNN is a convolutional or a vision transformer network; model complexity reflects the size of the network; visual diet reflects the number of images in the training set; and learning objective reflects whether the network was trained using a supervised, semi-supervised, or self-supervised learning objective. The full regression model contains all four factors. The plot showcases the extent to which the factors contribute to the variability observed in the performance of different models across a range of tasks and transformations. Asterisks indicate regression models that explained significantly more variance than a regression model with a constant term only.
Chapter 4

Discussion

4.1 DNNs are sensitive to 3D object transformations

Our examination uncovers a notable susceptibility of DNNs to 3D object transformations, resulting in a reduction in categorization performance as objects deviate from their canonical views. This trend aligns with recent progress in the realm of computer vision, encompassing investigations that have utilized both real-world photographic data and synthetically generated stimuli (Barbu et al., 2019; Alcorn et al., 2019; Abbas & Deny, 2022; Madan et al., 2021; Ruiz et al., 2022; Ibrahim et al., 2022).

While earlier research has generally revealed a consistent decrease in categorization performance under transformed conditions, making direct comparisons is complex due to multiple confounding factors. For instance, Abbas and Deny (2022) came remarkably close to achieving comparable outcomes, particularly in the context of object rotation. However, their study was marked by fewer degrees of variations, a smaller set of 3D object models (totaling 17), and evaluation across approximately 30 DNNs without meticulous control over the attributes of these networks.
In addition, the disparities in performance that we observe might also be partially attributed to our use of 16 distinct categories, in contrast to the conventional 1000 categories, leading to a narrower scope that potentially influences model behavior. Despite these differences, some patterns were replicated from previous research. For instance, the performance dips during object rotation along the x, and y axes (Figure 3.6, Figure 3.8, Figure 3.10) were mirrored in Abbas and Deny (2022), revealing shared challenges when dealing with specific unconventional angles such as those from the top or bottom of objects.

Curiously, when we expanded the visual diet of DNNs from 1.28 million images to 14 million images, we expected that greater diversity in images would mitigate the performance dips, leading to performance dips much like humans. However, as evident in Figure 3.8, this was not the case. We hypothesize that this could be due in part to the composition of DNNs’ visual diet, which lacks the naturalistic variety seen in humans. Currently, DNNs are trained on publicly available images, which do not fully encapsulate the distribution of naturalistic viewing conditions. These images tend to capture objects from typical viewpoints and lighting conditions, which might account for the observed disparities between DNN and human behavior.

Our investigation prompts intriguing comparisons between our findings and insights derived from the body of human developmental literature. Notably, research within human perception and cognitive development has proposed that the acquisition of invariance to object rotation in depth tends to mature more gradually compared to the invariance established for object translation or scale (A. J. Caron et al., 1979; Day & McKenzie, 1981; Gliga & Dehaene-Lambertz, 2007; Nishimura et al., 2015). Drawing inspiration from these developmental trajectories, we align our focus with the distinctive challenges posed by in-depth rotation for DNNs, which echo the observed developmental trends in humans.

Nevertheless, certain aspects of our results veer from the narratives delineated in the human literature. In the realm of scale transformations, DNNs exhibit a reduced level of performance, diverging from the expectations set by human perceptual processes. This divergence
could be attributed to the possibility that DNNs lean heavily on texture-based cues rather than shape cues [Geirhos et al., 2022], resulting in difficulties in accurately classifying smaller objects. These notable deviations underscore the intricate interplay between neural networks and human perception, elucidating both the shared traits and disparities in how they respond to the challenges posed by 3D object transformations.

4.2 Humans are more invariant than DNNs to object rotation in depth

In our exploration of the sensitivity of DNNs and humans to object rotation in depth, a distinct trend emerges, highlighting the superior invariance exhibited by human perception. This finding draws parallels with prior research on robustness, particularly the significant contributions of Geirhos et al. (2020). Their work, which investigated image-level rotation involving angles of 0, 90, 180, and 270 degrees, mirrors our examination of in-plane rotations (z-axis). Both studies consistently highlight that human observers exhibit greater resilience and robustness in the face of rotational transformations, further emphasizing the advanced perceptual capabilities of humans compared to DNNs.

Our findings also draw a relevant connection to recent research conducted by Lee and DiCarlo (2023). This study delves into a central challenge in visual object learning: the ability to accurately recognize new objects in novel images using a limited number of training images and across various 3D transformations (i.e. rotation, translation, background change, scale). While their work primarily focuses on human learning behavior, our investigation parallels their emphasis on the importance of appropriate learning rules. They employed a large set of testable learning models and conducted extensive psychophysics experiments involving novel 3D objects, revealing learning trajectories and behaviors in challenging, naturalistic settings. While these studies reveal DNNs’ advancements in approximating human performance, our
results indicate that humans still maintain an advantage across a spectrum of transformations.

The observed variations in performance among different DNNs in response to 3D object transformations provide intriguing insights into the factors influencing their robustness. Notably, our results suggest that increased training data, model complexity, and the incorporation of semi-supervised learning contribute to enhanced performance. This aligns with the notion that exposure to a larger and more diverse dataset can enable DNNs to encounter a broader range of object orientations, backgrounds, and sizes, thereby equipping them to handle unconventional scenarios more effectively.

The positive impact of additional training data on DNN performance is particularly noteworthy (Figure 3.7). DNNs trained on a richer visual diet of 14 million images consistently outperformed those trained on 1.28 million images across various transformation scenarios. This observation implies that the larger dataset exposes the models to a more comprehensive representation of object variations, allowing them to generalize better to novel situations.

The incorporation of semi-supervised learning also stands out as a significant factor contributing to enhanced performance (Figure 3.9). Semi-supervised models consistently exhibit favorable outcomes, often approaching or even surpassing human performance levels, particularly in reference views. This suggests that leveraging unlabeled data, in conjunction with labeled data, aids DNNs in capturing underlying patterns and variations that are crucial for robust object recognition.

### 4.3 Humans and DNNs make different errors

Our investigation not only highlights the discrepancy in error rates between humans and DNNs but also underscores the distinct nature of their error patterns. This distinction is particularly significant as it goes beyond the realm of error quantity and delves into the qualitative nature of errors. To ensure unbiased comparison, we employ an error consistency measure that
4.4 Visual diet and learning objectives are important for developing invariant object recognition

Our investigation emphasizes the vital roles played by visual diet and learning objectives in shaping DNNs’ ability to achieve invariant object recognition, particularly within the context of 3D object rotations.

When we analyze the findings from Figure 3.6, Figure 3.8, Figure 3.10 and Figure 3.13,
a clear theme emerges. These results highlight the significant influence of visual diet and learning objectives on a DNN’s robustness to object rotation. Importantly, our results reveal that larger models tend to benefit from exposure to a broader range of training images, especially when paired with semi-supervised learning approaches. This trend underscores the importance of employing diverse and comprehensive training data to foster improved performance across various transformations.

Furthermore, our observations align with prior research that has stressed the role of a comprehensive visual diet in shaping perceptual abilities. Earlier work by Geirhos et al. (2021); Smith et al. (2018); Conwell, Prince, Kay, Alvarez, and Konkle (2023) has drawn attention to the significance of offering neural networks a richer array of visual experiences, mirroring the developmental trajectory of human visual systems. This connection suggests that providing DNNs with a more realistic range of visual inputs can facilitate the development of robust object recognition capabilities.

Moreover, our findings resonate with studies that advocate for the effectiveness of self-supervised and semi-supervised learning strategies in enhancing network performance. The works of Konkle and Alvarez (2022); Zhuang et al. (2021) shed light on the potential benefits of leveraging these learning objectives, highlighting their capacity to promote more effective feature learning in neural networks.

In essence, our investigation underscores the pivotal roles of visual diet and learning objectives in the journey toward achieving invariant object recognition in DNNs. By thoughtfully curating training data and adopting suitable learning strategies, we can better equip artificial systems with the ability to perceive and recognize objects consistently across a range of transformations, echoing some facets of human visual cognition.
4.5 Limitations

While our study provides valuable insights into the behavior of humans and DNNs under various 3D object transformations, several limitations should be acknowledged. Firstly, the online nature of our experiment prevents us from closely observing participants during the task. However, the data quality, as indicated by the successful performance of the test block, suggests that the collected data is robust. Additionally, if necessary, future investigations could incorporate in-person testing with a smaller group of participants to validate the generalizability of the results.

Another limitation stems from the correlation among DNN dimensions within our model set. This correlation is evident from the regression results, where the removal of one dimension does not lead to a drastic drop in performance as anticipated based on the variance explained by individual model dimensions. This interdependence complicates the estimation of the individual contribution of each DNN dimension to the observed performance.

While our study addresses numerous aspects of the interplay between humans and DNNs in perceiving 3D object transformations, there remain avenues for future research to further enrich our understanding. Specifically, the following areas warrant further investigation:

- Conducting in-person experiments with a subset of participants to validate the generalizability of online findings.
- Exploring additional analysis techniques to untangle the contributions of correlated DNN dimensions and gain more granular insights into their impact.
- Extending the study to incorporate more diverse and unconventional object transformations, as well as investigating the impact of variations in lighting conditions and object textures.
- Exploring the integration of more recent DNN architectures and training methodologies
to assess their performance under similar conditions.

These future endeavors can address the current limitations and provide a more comprehensive understanding of the intricate relationship between human and DNN perception in the context of 3D object transformations.

4.6 Future work

While our study sheds light on the interaction between humans and DNNs in perceiving 3D object rotations, there are several avenues for future research that can expand and enhance our understanding of this complex phenomenon.

One prospective path involves expanding the scope of our investigations to encompass a wider spectrum of 3D object transformations beyond rotations. This extension can encompass object translation, changes in lighting conditions, variations in material or texture, and alterations in background, among others. Our current study, demonstrating the feasibility of online behavioral experiments, and paves the way for a comprehensive exploration of different transformations. The potential to replicate our findings across these transformations would underscore the broader applicability of human and DNN performance trends.

In-depth analysis of category confusion matrices presents another avenue for deeper understanding. By closely scrutinizing the distribution of errors within human participants and between humans and neural networks, we could gain insights into error consistency patterns and areas of divergence from human-like behavior.

Furthermore, delving into the internal representations within neural networks and linking them to error patterns offers a promising path. By investigating the effects of training distinct readouts from the final layer, we may unravel mechanisms underlying recognition processes, potentially leading to improved performance.
Given the critical role of training data quality, the exploration of visual diet characteristics becomes relevant. Future research can investigate the alignment between distributions of object orientations, sizes, and translations in training data (such as the widely used ImageNet dataset) and network performance on our test set, revealing insights into the impact of data characteristics on model performance. An intriguing avenue revolves around enhancing training data realism. The utilization of headcam data from infants, like the SayCAM (Sullivan, Mei, Perfors, Wojcik, & Frank, 2021) or Home View Corpus (Smith & Slone, 2017), offers a more naturalistic training regimen that mirrors human visual experiences during development. Evaluating networks trained on such data against human performance could offer novel perspectives on achieving human-like invariant object recognition through enriched training data.

Incorporating these future research directions can further elucidate the intricate dynamics between human and DNN perception of 3D object transformations and contribute to the development of more robust and human-like artificial visual systems.
References


Olshausen, B. A., & Field, D. J. (1996, June). Emergence of simple-cell receptive field properties
References


arXiv.1906.02168


Curriculum Vitae

Name: Haider Al-Tahan

Education:  
Honour B.Sc. in Computer Science and Psychology  
2014-2019  
York University  
MSc. in Computer Science  
2019-2021  
Western University

Awards:  
Vision Science Society travel award (2021)  
Ontario Graduate Scholarship (2021)  
Ontario Graduate Scholarship (2022)  
NSERC CGSD (2023)

Related Work:  
Teaching Assistant (2019 - 2021)

Experience:  
Western University  
AI Research Resident (2022 - Present)  
Meta AI
Publications:

