



Leveraging Human Brain Activity to Improve Object Classification

Citation

Fong, Ruth Catherine. 2015. Leveraging Human Brain Activity to Improve Object Classification. Bachelor's thesis, Harvard College.

Link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:14398538>

Terms of use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material (LAA), as set forth at

<https://harvardwiki.atlassian.net/wiki/external/NGY5NDE4ZjgzNTc5NDQzMGIzZWZhMGFIOWI2M2EwYTg>

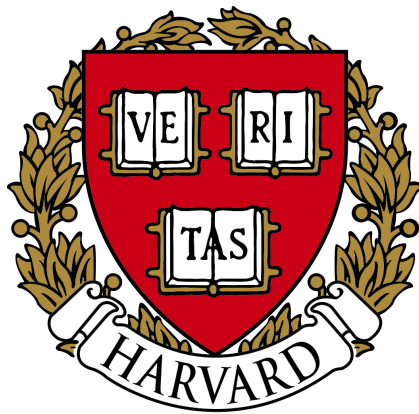
Accessibility

<https://accessibility.huit.harvard.edu/digital-accessibility-policy>

Share Your Story

The Harvard community has made this article openly available.
Please share how this access benefits you. [Submit a story](#)

Leveraging Human Brain Activity to Improve Object Classification



Ruth Catherine Fong
Department of Computer Science
Harvard College

April 1, 2015

“For it is God who works in you, both to will and to work for his
good pleasure.” - Philippians 2:13

Acknowledgements

First, I would like to thank my advisor Professor David Cox for his generous support over the past year. Your willingness to patiently explain nearly two decades worth of neuroimaging research late into the night barely captures the depth of your mentorship. My future ambitions would have been wholly different had you not seeded and continued to nurture my curiosity in this work. I would also like to express my deepest appreciation to my research mentor Dr. Walter Scheirer for his constant patience and encouragement. From your readiness to help debug my code to our many meetings in which you motivate me to press on, I am indebted to kindness and advising. I would like to thank both Professor Cox and Dr. Scheirer for working closely with me to prepare a paper for submission based on this research. I am thankful for the rest of the Cox lab, who have helped read many of my applications, answered countless questions, and contributed to a truly enjoyable lab culture.

Additionally, I would like to thank Professors Ryan Adams and Todd Zickler for agreeing to read my thesis and first inspiring me in the areas of machine learning and computer vision through their courses, CS181 and CS273. I am also grateful to Professor Harry Lewis for his mentorship in teaching and life. Thank you for giving me opportunities to experience the joys (and pains) of higher education from the other side.

Next, I would like to thank my main CS partner, Emily Lawton, and fellow teaching fellows, Louis Li and Yifan Wu. Thank you Emily for the utmost joy the past two years have been working together and for putting up with my tardiness; I could not imagine anyone else with

whom I would want to take nearly half of my courses. I am grateful to both Louis and Yifan for their encouragement in my teaching and research potential, and for the many times we spurred one another on to be better educators, learners, and scientists.

I also wanted to thank the lovely ladies of the 9th floor of Mather: Meredith Arra, Rosa Huang, Chloe Reichel, and Carrie Tian. A wall of words could not describe the many ways my roommates have contributed to this thesis and, more importantly, to my growth as a person. Thank you for being my thesis gnomes, karaoke companions, fashion advisors, and the village that raised this girl.

I am also grateful to Marina Chen, Shannon Fan, and Jenny Shih, for their company in my last days of thesis-ing; thank you for putting up with my preference for dungeon-like computer labs. In addition, I am thankful for Brandon Li and the rest of the Harvard-Radcliffe Asian-American Christian Fellowship for encouraging and filling in for me as I scaled back in my involvement to work on this thesis.

I am indebted to the many people who severely reduced the number of typos and awkward sentences in this thesis, which includes some mentioned early, as well as Stella Pantella, Kevin Lee, Alisa Nguyen, Stephen Mackereth, Tom Silver, Stella Tu, Jennifer Tu, Travis Yeh, Carolyn Killea, and Valentina Perez. Thank you for taking the time to correct subject-verb agreements and egregious spelling errors.

Finally, I would like to thank some of my most fervent supporters. Thank you Brian for your selfless love, not just during the thesis crunch, but every day for the past 1.57 years. I am grateful for our mutual growth and encouragement, and I am hopeful for many more such moments. I am also hugely indebted to my parents, Drs. Tung and Chin-lin Fong, who have always believed in me more than I believed in myself and who nurtured in me an insatiable curiosity for learning. Thank you for your many sacrifices for my future and well-being. Lastly, I am eternally grateful to my Maker. May I live day-by-day for your will will be done and not my own. *Soli Deo Gloria.*

Abstract

Today, most object detection algorithms differ drastically from how humans tackle visual problems. In this thesis, I present a new paradigm for improving machine vision algorithms by designing them to better mimic how humans approach these tasks. Specifically, I demonstrate how human brain activity from functional magnetic resonance imaging (fMRI) can be leveraged to improve object classification.

Inspired by the graduated manner in which humans learn, I present a novel algorithm that simulates learning in a similar fashion by more aggressively penalizing the misclassification of certain training datum. I propose a method to learn annotations that capture the difficulty of detecting an object in an image from auxilliary brain activity data. I then demonstrate how to leverage these annotations by using a modified definition of Support Vector Machines (SVMs) that uses these annotations to weight training data in an object classification task. An experimental comparison between my procedure and a parallel control shows that my techniques provide significant improvements in object classification. In particular, my protocol empirically halved the gap in classification accuracy between SVM classifiers that used state-of-the-art, yet computationally intensive convolutional neural net (CNN) features and those that used out-of-thebox, efficient histogram of oriented gradients (HOG) descriptors. Further analysis demonstrates that my experimental results support findings in neuroimaging literature about the roles different cortical regions play in object recognition.

Contents

Contents	v
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 Overview	1
1.2 Contributions	4
1.3 Related Work	4
1.4 Outline	8
2 Preliminaries	9
2.1 Functional Magnetic Resonance Imaging	9
2.1.1 Regions of Interest (ROIs)	11
2.2 Perceptual Annotation	13
2.2.1 Binary Classification	13
2.2.2 Loss Functions	14
2.3 Support Vector Machines (SVMs)	17
2.3.1 Platt Probabilities	21
2.4 Image Feature Representation	22
2.4.1 Histogram of Oriented Gradients (HOG)	22
2.4.2 Convolutional Neural Networks (CNN)	24
3 Experimental Design and Methods	28
3.1 Data	28

3.1.1	Dataset	28
3.1.2	fMRI Data Preprocessing	29
3.1.3	Image Pre-processing	30
3.1.4	Label Assignment	30
3.1.5	Image Features	31
3.2	Experimental Design	32
3.2.1	Data Partitions	32
3.2.2	Experiments	32
3.2.3	Generating Perceptual Annotations from Brain Activity .	33
3.2.4	Balanced Classification Problem	36
4	Results and Analysis	38
4.1	Experimental Set-up	38
4.1.1	Experiments	38
4.1.2	Simplifying Assumption	39
4.1.3	Note about VEHICLES Category	39
4.2	Basic Statistics	39
4.2.1	Significance Testing	40
4.3	Basic Results	40
4.3.1	Comparison of HOG vs. CNN Image Features	43
4.4	Results by Combinations of ROIs	43
4.4.1	Feature-Level Fusion of ROIs	43
4.4.2	Top Combinations of ROIs	44
4.5	Analysis of ROIs	49
4.5.1	Null Distributions	51
4.5.2	Bonferroni Confidence Intervals	54
4.6	Comparison of Different Perceptual Annotation Weighting Schemes	57
5	Discussion	60
5.1	fMRI-based Perceptual Annotations	60
5.1.1	Extension: Multi-dimensional Perceptual Annotations . . .	61
5.1.2	Extension: Generating Perceptual Annotations from Other Sources	61

CONTENTS

5.2	Moving towards state-of-the-art, CNN-like performance with HOG	63
5.2.1	Extension: CNN-based Perceptual Annotations for HOG)	64
5.3	Other Decision-Making Algorithms	64
6	Conclusion	66
	Appendix A	68
	Appendix B	70
	References	73

List of Figures

2.1	Voxels in the Brain	11
2.2	Regions of Interest (ROIs)	12
2.3	Examples of Good and Bad Margins for SVMs.	18
2.4	Histogram of Oriented Gradients (HOG) Workflow.	22
2.5	Example of HOG Descriptor.	23
2.6	Visualization of Neural Network Features.	25
2.7	Network Diagram	27
3.1	Recording Brain Activity	29
3.2	Example of Generated HOG Features	31
3.3	Example of Generated CNN Features	32
3.4	Experiment Design	33
3.5	Generating Platt Probability Perceptual Annotations From Brain Activity	35
3.6	Re-Weighting Training Samples Using Perceptual Annotations . .	37
4.1	Basic Side-by-Side Comparison of Results	41
4.2	Significant Improvements in Most ROI Combinations	42
4.3	Results from Individual ROIs: EBA, FFA, and PPA	45
4.4	Comparison of Combinations With EBA	46
4.5	Top Five Combinations of ROIs for HUMANS	47
4.6	Top Five Combinations of ROIs for ANIMALS	47
4.7	Top Five Combinations of ROIs for BUILDINGS	48
4.8	Top Five Combinations of ROIs for FOODS	48
4.9	Influence of Individual ROIs	50

LIST OF FIGURES

4.10 Null Distributions for Analysis of ROIs (HOG)	52
4.11 Null Distributions for Analysis of ROIs (CNN)	53
4.12 Analysis of ROIs (HOG)	55
4.13 Analysis of ROIs (CNN)	56
4.14 Number of Significance ROI Combinations using Direct and Trans- formed Platt Annotations	58
4.15 Results Comparison Between Direct and Transformed Platt Anno- tations	59

List of Tables

3.1	Regions of Interest (ROIs) in Early and Later Visual Cortex . . .	29
3.2	Number of Images in Each Object Category	31
4.1	Count of ROIs Included in the Top Five Combinations for each Object Category.	49

Chapter 1

Introduction

How can computers think and act more like humans? In the past few decades, computer algorithms have beaten humans at well-defined games like chess and Jeopardy (Campbell et al., 2002; Ferrucci et al., 2010). Yet, they still perform poorly on intuitive tasks that humans excel at subconsciously, such as recognizing different kinds of animals. The best, state-of-the-art object detection algorithms perform with an average precision rate between 30-40% (Russakovsky et al., 2014). Yet, toddlers easily and excitedly learn to identify all sorts of critters quite well. Today, the most challenging problems in artificial intelligence (AI)—reading body language, understanding humor—typically share the quality of being trivial tasks for humans.

1.1 Overview

This defining characteristic underlies the overarching goal of AI to close the gap between human and computer performance in a variety of tasks and motivates the recent trend of biologically-inspired approaches to many such problems (Cadieu et al., 2014; Cox and Pinto, 2011; Pinto et al., 2009). However, biologically-inspired algorithms in machine learning have focused more on representating information in biologically-plausible ways than on making decisions in a more human-like way. In this thesis, I present a new paradigm for improving machine learning algorithms by designing them to better mimic how humans tackle these

tasks.

So, how *do* humans learn? Consider a five-year-old child learning math. She may be able to learn simple addition and subtraction. Yet, would she be able to solve algebraic expressions? Or, could she formally prove that $\sqrt{2}$ is irrational? This progression illustrates the concept of “learnability”: Some concepts are not as “learnable” as others. It also demonstrates the notion of graduated learning. Humans typically learn easier things before more difficult ones, and we often generalize our easier experiences to tackle harder challenges. After mastering basic arithmetic operations, our young pupil may be able to learn algebra and geometry. However, we would never ask her to master the whole breadth of the field of mathematics (this task is impossible for most individuals of all ages). Similarly, “Never-Ending Language Learning” (NELL), a continuously growing knowledge base designed to mimic how humans learn throughout their lives, also learned “easier” concepts first (Carlson et al., 2010).

While Valiant (1984) formalizes the similar concept to “learnability”, most machine learning algorithms take a sink-or-swim approach. Consider the task of creating a cat detector that can identify (and localize) the presence of a feline in an image. To build such detector, algorithms would be provided labelled “cat” and “no cat” images to learn from. An image in which a feline is the central, dominant object may be more “learnable” than one in which a cat is partially obscured by the presence of many other creatures. Yet, most machine learning techniques would not consider the “learnability” of each cat photo.

Motivated by the idea that algorithms should prefer solutions that are more likely to be chosen by humans, Scheirer et al. (2014) introduces a technique called perceptual annotation and presents empirical results that demonstrate its ability to outperform state-of-the-art face detection algorithms. The algorithm prioritizes correctly detecting faces that humans quickly and easily identify and worries less about correctly detecting the faces that even humans struggle to recognize consistently. It accomplishes this through a modified definition of the popular machine learning algorithm Support Vector Machines (SVMs).

This thesis extends Scheirer et al. (2014) in two significant ways. First, to generate annotations that, in part, capture the “learnability” of a training datum, I leverage rich, highly-dimensional brain activity data from function mag-

netic resonance imaging (fMRI) scans. In contrast, [Scheirer et al. \(2014\)](#) used two-dimensional visual psychophysics data that captured the response time and overall accuracy of humans performing face detection tasks. Second, I apply the concept of perceptual annotations to the different space of object recognition. Whereas [Scheirer et al. \(2014\)](#) applied perceptual annotations to face detection, I focused on extending the principle to the arguably more challenging task of object classification.

Due to the sparsity of image data and the high variability of objects, object recognition continues to pose a formidable challenge to the machine learning and computer vision communities. Thanks to the advent of Facebook, the amount of high-quality, labelled face images with rich social network information ([Stone et al., 2008](#)) has exponentially increased in the past decade. The increased depth and scale of face data has narrowed the gap between human and computer performance in face detection. While the computer vision community has continually been developing high-quality object datasets ([Lin et al., 2014](#); [Russakovsky et al., 2014](#)), they still pale in comparison to the richness of face data. More fundamentally, object recognition suffers from the fact that objects are incredibly diverse and variable. Within a single object class, such as CARS or CATS, there often exists large intra-class variance in how objects appear, as evidenced by the diverse set of car models today and the wide variety of feline breeds. Moreover, when considering the wide variety of natural scenes, objects can be plausibly co-located with other objects in countless ways. In contrast, faces often appear in images, like ones of social gatherings, that are naturally constrained by setting.

Yet, patterns in brain activity observed in humans viewing natural images may help in tackling these two significant challenges in object recognition. The densely rich information encoded in brain activity may help offset the relative lack of image data for objects. Neuroimaging data also co-opts years of human experience navigating the world. Photographs barely capture the deep wealth of human understanding about the world, yet this encoded knowledge may help constrain how objects naturally appear or are co-located with other things in scenes.

1.2 Contributions

The overarching contribution of this thesis is to establish that brain activity data can be used to substantially narrow the gap between human and computer performance in challenging AI tasks. The main contributions of this thesis are as follows.

- I demonstrate how to leverage a previously untapped yet existing source of information —rich, highly-dimensional brain activity data from fMRI scans—to augment machine learning and computer vision algorithms.
- I present a novel, two-stage classification algorithm that takes a biologically-inspired approach to decision making. First, I present an original technique to reduce high-dimensional brain activity to salient, one-dimensional annotations. Second, I outline how to use a modified SVM classifier to more aggressively penalize the misclassification of certain training datum based on brain activity annotations.
- I present a case study in object classification that highlights how my biologically-informed algorithm significantly improves classification accuracy compared to a standard SVM classifier that uses a hinge loss function.

1.3 Related Work

In the past decade, much state-of-the-art work in machine learning and computer vision has come from models that attempt to represent knowledge in biologically-consistent ways. Inspired by work in [Hubel and Wiesel \(1963\)](#) on how the early visual cortex uses feed-forward processing, [LeCun et al. \(1998\)](#) presented convolutional neural nets (CNNs) —a model that mimics the receptive fields of neurons and their ability to withstand geometric variation in visual stimuli. Since then, continual research has been done to better simulate the hierarchical, feed-forward mechanisms of primate visual systems ([Serre et al., 2007b,c](#)), particularly higher-level ventral stream regions ([Serre et al., 2007a](#)), like the inferior temporal (IT) cortex ([Riesenhuber and Poggio, 1999](#)), that are responsible for understanding

semantic meaning. Recent work demonstrates strong correlations and representational similarities between the best models of the primate IT cortex and high-performing neural networks in various object recognition tasks (Cadieu et al., 2014; Yamins et al., 2013, 2014). While many models focus on the heavily-studied visual system, computational models of the full brain have also been developed that can coordinate complex tasks like viewing and copying text using a mechanical arm (Eliasmith, 2013; Eliasmith et al., 2012).

Hierarchical, feed-forward models are most often used to generate state-of-the-art, biologically-inspired features to be used in tasks like object recognition (Serre et al., 2005). These features appear to capture higher-level image characteristics compared to other descriptors like Histogram of Oriented Gradients (HOGs) generated from low-level, geometric properties (Dalal and Triggs, 2005). Thus, they are believed to perform better in tasks involving semantic understanding because they better mimic the visual regions responsible for higher-level processing, while features such as HOG or SIFT features (Lowe, 1999) better parallel the early visual cortex’s low-level processing of qualities like edges. Much research has also been invested in choosing the best parameters for biologically-inspired features. To simulate the highly parallel nature of the visual systems, GPU processors perform a data-driven search of the parameter space in order to find settings that consistently perform well in a variety of recognition tasks (Bergstra et al., 2013; Cox and Pinto, 2011; Pinto et al., 2009; Vig et al., 2014). Fei-Fei and Perona (2005) and Li et al. (2010) describe biologically-inspired features that do not attempt to model the visual system explicitly but nonetheless focus on characterizing images semantically by using descriptors from object banks instead of texture banks. Most recently, a reinforcement learning algorithm that uses CNNs learned how to play the classic 1980s game Atari at a level comparable to competitive gamers (Mnih et al., 2015).

While biologically-inspired work in machine learning has focused on knowledge representation, major breakthroughs in machine learning have come in the form of ensemble techniques, which mimic the parallel nature of human cognition. Similar to the high-throughput screening of biologically-inspired features (Vig et al., 2014), ensembles work by combining many weak learners, typically decision trees (Quinlan, 1986), to form a strong one. One such technique called

AdaBoost trains several weak learners and combines them as a weighted sum into a final classifier (Freund and Schapire, 1995). Random forests work similarly in that they train several decision trees and output the mode or mean prediction for classification or regression tasks respectively (Breiman, 2001). When applied to image classification tasks, random forests perform significantly better than SVMs (Bosch et al., 2007). These techniques correct the overfitting that decision trees often induce by relying on redundancy, a basic principle that also underpins how neurons work together (Miller, 1982). In addition to ensemble methods, graduated learning is another biologically-inspired idea that has motivated classic machine learning techniques like expectation maximization (Dempster et al., 1977) and gradient descent, both of which focus on making learning a model in incremental steps.

Much research has also been done on the use of annotations or auxiliary data in machine learning and computer vision algorithms. Welinder et al. (2010) demonstrates how using crowd-sourced annotations for object labels conveys rich information about images and can be used directly in classifiers to compete with state-of-the-art recognition algorithms. In a similar vein of using crowd-sourced annotations for classification, Tamuz et al. (2011) demonstrates how a similarity matrix “crowd kernel” using only crowd-sourced annotations can be learned; when such a matrix is used as a kernel for an SVM classifier, obvious and subtle, distinctly human-like distinctions are learned, such whether neckties have “stripes” and whether letters are vowels or consonants. Branson et al. (2011) establishes a framework for automatically generating annotations of where parts are on an object (i.e. where the head, beak, and tail are each located in a photo of a bird) by alternating between interactive learning that solicits human annotations and online learning. Perceptual annotations are similar to this work in that they represent an attempt to augment algorithms with human annotations. Leveraging context in scenes to improve object recognition has also been well-researched (Li and Fei-Fei, 2007; Torralba et al., 2006). Similarly, Stone et al. (2008) demonstrates how social network information can be used to augment face recognition in Facebook photos, while Gardner et al. (2013) demonstrates how adding more descriptive, “lexicalized” labels to edges in a graph of a knowledge base improves its performance on inference tasks by reducing data sparsity. De-

spite the body of research on annotations, particularly crowd-sourced ones, little work has been done to use brain activity as a source of annotations.

Machine learning and computer vision research with brain activity data from fMRI scans has been focused on four main problems: 1. predicting fMRI activity, 2. using brain activity data directly for computer vision tasks, 3. “decoding” fMRI data to better understand how different cortical regions work, and 4. relating different modes of cognitive data to one another. [Mitchell et al. \(2008\)](#) demonstrates how brain activity recorded while a human reads a noun can be predicted for hundreds of nouns by training a model on fMRI scans of when humans view several dozen nouns. [Stansbury et al. \(2013\)](#) also demonstrates how a model can be trained to predict the brain activity of humans when they view natural images of scenes.

Other research illustrates that models trained on fMRI data can predict which image is being viewed from a fixed set of images ([Kay et al., 2008](#)) as well as to which category that image belongs ([Shinkareva et al., 2008](#)). Similar work has shown that models can be trained on brain activity to reconstruct images ([Naselaris et al., 2009](#)) and videos ([Nishimoto et al., 2011](#)) that are similar to the images and videos presented to subjects during their fMRI scans.

Decoding the brain has been a highly active research area at the intersection of machine learning and neuroscience, with much research in this space using multivariate analysis to extract, detect, and classify salient patterns about cortical functions from brain activity ([Cox and Savoy, 2003](#); [Mur et al., 2009](#); [Stansbury et al., 2013](#)). Extending such research, [Reddy et al. \(2010\)](#) demonstrates that decoding techniques produce similar results when used on brain activity of humans viewing visual stimuli compared to activity of humans imagining visual stimuli. [Huth et al. \(2012\)](#) also builds on and synthesizes classical decoding work by presenting an interactive map of the brain that illustrates how different regions are activated by various objects and actions.

One final area of research at the junction of neuroimaging and machine learning focuses on relating similarities between different kinds of cognitive data. Interested in relating brain activity data, behavioral data, and computational models of different cortical regions to one another, [Kriegeskorte et al. \(2008a\)](#) presents a modality-independent representation of information called representational dis-

similarity matrices (RDMs) and a method for comparing such representations of data collected through different modalities. Similarly, [Kriegeskorte et al. \(2008b\)](#) observes similarities in how the IT cortex of humans and monkeys functions in object recognition by using RDMs to compare fMRI scans of humans to single-electrode recordings of monkeys.

1.4 Outline

Building off the many ideas and findings presented in the previous section, this thesis presents a novel mechanism to generate annotations from brain activity that improves the decision-making aspect of object classification. In the next chapter (Chapter 2), I introduce relevant background on fMRI data, SVMs, and image features, and I also describe the modified SVM formulation that incorporates perceptual annotations. I describe my dataset, experimental design, and methods in Chapter 3. In Chapter 4, I present experimental results and analysis that demonstrate the efficacy of my algorithm in improving object classification performance as well as highlight which cortical regions significantly contributed to boosting accuracy. Chapter 5 discusses the implications of this research and presents ideas for future work. Finally, Chapter 6 concludes.

Chapter 2

Preliminaries

This chapter provides technical background for several major concepts used throughout this thesis.

2.1 Functional Magnetic Resonance Imaging

Functional Magnetic Resonance Imaging (fMRI) is used to detect brain activity by measuring the changes in blood oxygenation and blood flow in the brain. In this section, I provide a brief overview on the development of fMRI technology and describe how fMRI measure brain activity. For more details about fMRI, see [Huettel et al. \(2004\)](#) and [Buxton \(2009\)](#).

Magnetic resonance refers to a property discovered by physicist Isidor Isaac Rabi. In the early 1930s, Rabi described how atomic nuclei “flipped” under the combined influence of a strong, external magnetic field and radio waves ([Watson, 2011](#)). Typically, nuclei point in random directions, but when hit with a magnetic field and radio waves, they are disturbed and briefly align themselves more with the direction of the field ([Devlin, 2008](#)). As the disturbed nuclei “snap back” into random orientations, they emit a magnetic signal that can be measured. The strength of this signal varies based on the surrounding material, such as gray matter or hemoglobin (blood). This insight led to the development of Magnetic Resonance Imaging (MRI), which generates detailed, static images of biological structures —typically organs and arteries —by measuring their magnetic reso-

nance. fMRI was developed in the 1990s, when physicist Seiji Ogawa discovered that oxygenated or oxygen-rich blood from the lungs possessed different magnetic properties than deoxygenated or oxygen-poor blood that was returning to the lungs (Watson, 2011). While Ogawa is credited with this discovery, Linus Pauling suggested this idea in the 1930s. Regardless, this insight allows MRI technology to measure changes in blood flow and oxygenation levels in order to quantify brain activity.

Although fMRI scans do not directly measure neuronal activity, they detect brain activity by relying on the fact that increased neuronal activity in a given area of the brain leads to increased blood flow and the recruitment of oxygenated blood. Because oxygenated and deoxygenated blood have different magnetic properties, they possess different levels of magnetic resonance. This allows the blood oxygenation level dependent (BOLD) contrast, or the relative changes in blood flow and oxygenation, to be measured by fMRI. In reality, neuronal activity does not immediately engender an influx of oxygenated blood to a cortical region (Devlin, 2008). Instead, blood levels typically dip right at the beginning of increased neuronal activity, as energy in the form of sugar is needed to support such activation. After the “initial dip,” the brain overcompensates by recruiting an overabundant influx of oxygenated blood, leading to high, oxygenated blood levels. Blood levels typically peak six seconds after the initial increased neuronal activity. Finally, blood levels drop back down to baseline levels, often with another dip called “post-stimulus undershoot,” before returning to back to normal.

fMRI scans record brain activity using units called voxels (Fig. 2.1). A voxel is a three-dimensional unit in space and is analogous to a pixel in images. Each voxel includes activity from thousands of neurons (Watson, 2011). A response amplitude value is given to voxels in order to generate images of brain activity like the one in Fig. 2.2. To calculate this value, the timecourse of a voxel’s BOLD signals is compared to the dip-peak-dip pattern expected of increased neuronal activity (Devlin, 2008). Strong correlation results in a large, positive response amplitude value while inverse correlation yields a negative value.

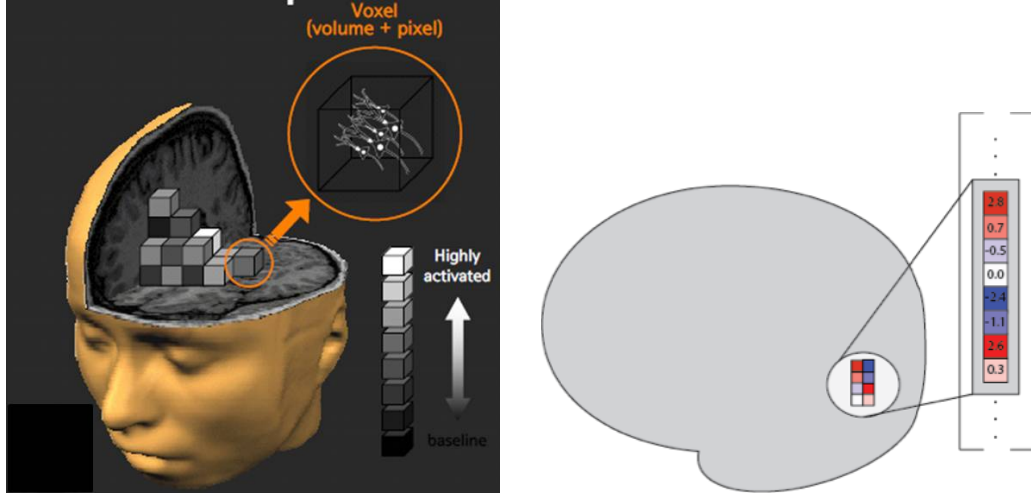


Figure 2.1: Voxels in the brain. A voxel is a three-dimensional unit and is analogous to a pixel in images. In fMRI, voxels are the basic units that have activation values. The left image first appeared in [Park \(2011\)](#).

2.1.1 Regions of Interest (ROIs)

Often, functional regions of interest (ROIs) are identified in fMRI scans in order to specifically analyze the voxel activity in that area (Fig. 2.2). The most common way to draw the anatomical boundaries of ROIs is by using a “localizer” to identify these functionally-specific areas in a scan that is separate from the ones that are being used for research ([Poldrack, 2007](#)). For instance, to determine the location of the fusiform face area (FFA), a participant is shown images of faces, objects, scrambled faces, and scrambled objects. Using response amplitude values, the FFA is identified to be the area in which voxels are highly active for faces but not the other three categories of things.

In this thesis, I focus on using brain activity from the following 7 ROIs:

1. **Extrastriate body area (EBA)**, which is involved in recognizing the human body and body parts ([Downing et al., 2001](#)).
2. **Fusiform face area (FFA)**, which aids in face recognition ([Kanwisher et al., 1997](#)), although some research suggests that FFA may select for familiar items, such as cars and birds ([Gauthier et al., 2000a](#)).

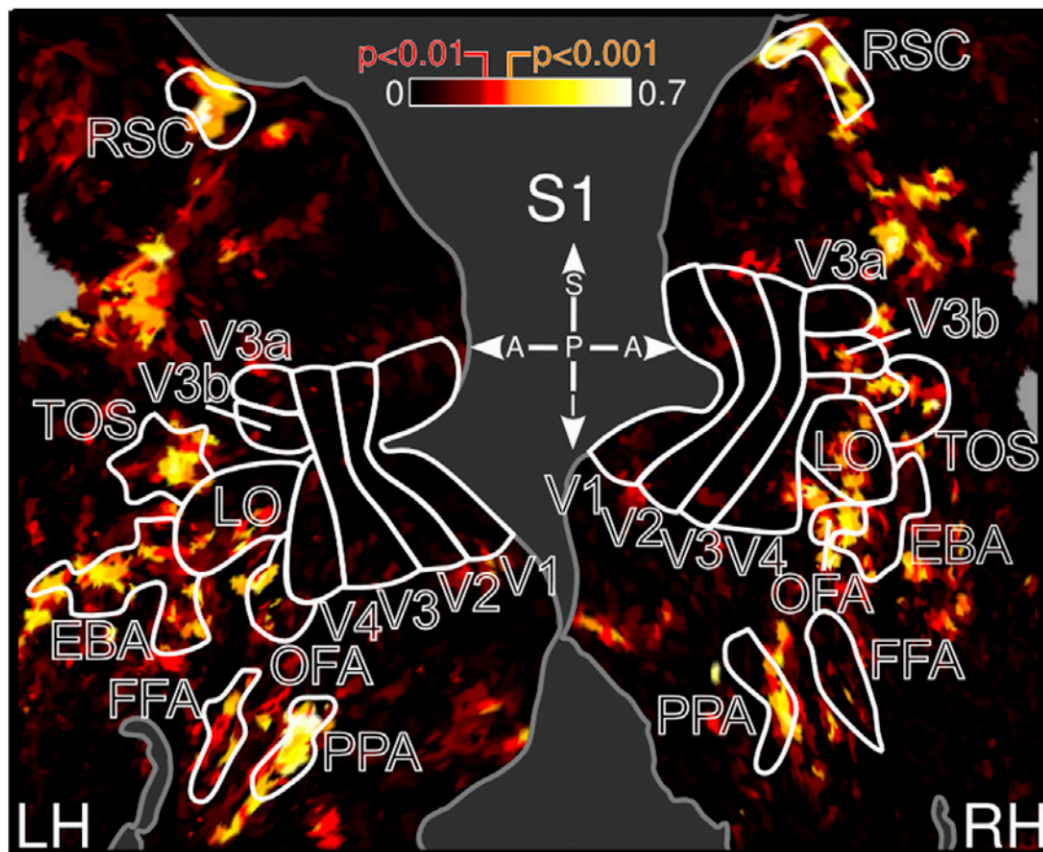


Figure 2.2: Regions of Interest (ROIs). ROIs are outlined in white and are known to be responsible for various high-level visual functions. This figure is Figure 3a in [Stansbury et al. \(2013\)](#).

3. **Lateral occipital cortex (LO)**, which is important for object recognition (Grill-Spector et al., 2001; Malach et al., 1995).
4. **Occipital face area (OFA)**, which works with FFA in face recognition by detecting individual parts of a face, such as a nose or a pair of eyes (Gauthier et al., 2000b).
5. **Parahippocampal place area (PPA)**, which is most active when an individual is viewing scenes or “places” (Epstein and Kanwisher, 1998).
6. **Retrosplenial cortex (RSC)**, which is important for human navigation tasks and aids in episodic memory (Maguire, 2001).
7. **Transverse occipital sulcus (TOS)**, which is activated by visual stimuli that includes buildings (Gauthier et al., 2000b; Hasson et al., 2003; Nakamura et al., 2000).

2.2 Perceptual Annotation

Scheirer et al. (2014) introduces the concept of perceptual annotation as additional information on training data. Each perceptual annotation is a single, scalar value that captures the “learnability” of an image in a visual task, such as face recognition or object classification. Scheirer et al. (2014) outlines a modified Support Vector Machine (SVM) that uses a novel “human-weighted” loss function to more aggressively penalize the misclassification of certain training data based on their annotations.

In this section, I will briefly explain the problem of binary classification before presenting the perceptual annotation approach introduced in Scheirer et al. (2014).

2.2.1 Binary Classification

In this thesis, I present results that demonstrate how perceptual annotations significantly improve the accuracy rates of binary classification problems for several object categories, such as HUMANS.

Binary classification refers to the problem of learning to predict a label $p \in \{+1, -1\}$ for a data point $x \in \mathbb{R}^d$. Let a predicted label be given by a classification function, $p = f(x)$, where $f : \mathbb{R}^d \rightarrow \{+1, -1\}$. Then, binary classification becomes focused on finding a classification function f from a set of training data $\{(x, y) \in \mathbb{R}^d \times \{+1, -1\}\}$. This function f can then be used to predict the labels of test data. Typically, no learnable and generalizable function f exists in which all the predicted labels $f(x)$ correspond with each data point's target label y . Thus, classification becomes focused on finding the function f that best minimizes incorrect predictions. More formally, the task aims to identify a classification function f that minimizes ideal risk given by R_I (Smola and Schölkopf, 1998):

$$\arg \min_f \{R_I(f) := \int_{\mathbb{R}^d \times \mathbb{N}} l(x, y, f(x)) P(x, y)\} \quad (2.1)$$

Ideal risk R_I contains two terms: the loss function $l(x, y, f(x))$, which assigns a penalty for misclassification, and the joint distribution $P(x, y)$ of data points x and their target labels y .

Let's consider an example. In the case of classifying 100×100 images based on whether or not they contain HUMANS, a data point $x \in \mathbb{R}^{10000}$ is the pixel values of the image stored as a vector, and a target label $y = +1$ would denote the associated image as a positive HUMANS sample, while $y = -1$ would denote a negative sample that does not contain HUMANS.

2.2.2 Loss Functions

In statistical learning, loss functions are used to assign penalties for misclassifying data. Scheirer et al. (2014) implements the concept of perceptual annotation through a modified loss function for SVMs to weight the penalties of training data based on their perceptual annotations.

Typically, SVMs use a hinge loss (HL) function. Let $z = y \cdot f(x)$ denote the correctness of a prediction, as $z = 1$ when $f(x) = y$ and $z = -1$ otherwise. Then,

the hinge loss function is as follows:

$$\zeta_h(z) = \max(0, 1 - z) \quad (2.2)$$

The HL function assigns a uniform penalty to all misclassified data. However, our intuition about “learnability” suggests that not all training samples are created equal. Perceptual annotations aim to better reflect differences in “learnability” among training data. To do this, [Scheirer et al. \(2014\)](#) introduces an original “human-weighted” loss (HWL) function. Before discussing HWL, let’s discuss the setting of perceptual annotations.

Consider a set of L training samples comprised of two subsets: one with perceptual annotations and one without. This setting provides flexibility by not requiring all training data to be perceptually annotated, which is useful in cases where generating perceptual annotations for all training data is difficult. This is the case with perceptual annotations based on neuroimaging data.

Let the set of m perceptually annotated training examples be denoted by $P = (x_i, y_i, c_i)_{i=1\dots m}$, where $x_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$, and a cost $c_i \in \mathbb{R}$. Similarly, let the set of n typical training examples be denoted by $T = (x_j, y_j, c_j)_{j=1\dots n}$. Then, our training set $X = P \cup T$ and $|X| = m + n = L$.

Then, HWL can be formalized with a function $M : \mathbb{R}^d \rightarrow \mathbb{R}$ that maps each data point x_i to its cost c_i :

$$\zeta_\psi(x_i, z_i) = \max(0, (1 - z_i) + M(x_i, z_i)) \quad (2.3)$$

where

$$M(x_i, z_i) = \begin{cases} c_i, & \text{if } z_i < 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.4)$$

This definition allows an additive penalty to be incurred by the $M(x_i, z_i)$ term in the case of misclassification. The cost value c_i can either be a static penalty or vary based on training sample x_i and is where perceptual annotations come in. For the perceptually annotated examples $x_i \in P$, c_i takes on the value of an annotation generated from auxillary data such as fMRI brain activity that captures the “learnability” of the example. In contrast, for typical examples

$x_i \in T$, c_i takes on a static penalty that is less than the minimum perceptual annotation in P . In the experiments described in chapter 4, I fixed cost $c_i = 0$ for all the typical, non-annotated examples. For these samples, ζ_ψ reduces to the hinge loss function. This formulation chooses solutions that are more like the ones humans would choose, as it more heavily penalizes the misclassification of perceptually annotated samples based on their “learnability” by humans.

In SVMs, the basic classification function is defined as follows:

$$f(x) = \begin{cases} +1 & \text{if } w^T \cdot \phi(x) + b > 1 \\ -1 & \text{otherwise} \end{cases} \quad (2.5)$$

where w is the weight vector and b is the bias term, both of which are learned by the model, and $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^j$ is a function that projects an example into higher-dimensional space for better feature representation (SVMs will be explained in more detail in section 2.3). Then, assuming the training data is linearly separable, that is, that a function f exists that correctly classifies all training data, we can find a classification function f by solving the following optimization problem:

$$\min \left\{ \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^L \zeta_\psi(x_i, y_i, f(x_i)) \right\} \quad (2.6)$$

where the C parameter determines how much to weight each term in Equation 2.6, where the first term maximizes the margin (a critical concept in SVMs) and the second term minimizes misclassification in the training data.

One interesting facet of Equation 2.6 is that it is not convex. While convexity is often valued in machine learning because it guarantees a globally optimal solution, it is not consistent with how biological visual systems work and can select overly simple solutions for complex, holistic visual tasks like recognition. DiCarlo and Cox (2007) demonstrate that such systems are made up of many non-linear, adaptive layers, while the state-of-the-art performance of the convolutional neural networks, which are not convex, suggests that the best solutions may be found by biologically-inspired, non-convex algorithms. Not only is the non-convex formulation of human-weighted loss more biologically-consistent, but it also does not pose any practical limitations in terms of computation time. Collobert et al.

(2006) shows that non-convex loss functions for SVMs reduce training time and space constraints. For this thesis, no significant difference in computation time was observed between experiments that used the convex, hinge loss function and those that used the non-convex, human-weighted loss function.

2.3 Support Vector Machines (SVMs)

Because Scheirer et al. (2014) implements perceptual annotation with a modification to SVMs, this section will provide a concise overview of support vector machines, a popular algorithm for supervised learning, which refers to machine learning tasks in which the target labels are provided in the training set. This section is based on Pfeffer et al. (2014) and Gavrilov (2015). For a more thorough treatment and derivation of SVMs, see Chapter 7 in Bishop et al. (2006).

First, let's consider the linearly separable case. Then, SVMs aim to find a hyperplane or decision boundary that divides positive and negative data; the boundary will have the following equation:

$$w^T \phi(x) + b = 0 \quad (2.7)$$

where the weight vector $w \in \mathbb{R}^j$ and the bias value $b \in \mathbb{R}$ are parameters of the SVM to be learned and $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^j$ is a collection of j basis functions that produces a useful feature representation of $x \in \mathbb{R}^d$ in an often higher-dimensional space $\mathbb{R}^j$. Then, the classification function is given by the function $f : \mathbb{R}^d \rightarrow \{+1, -1\}$:

$$f(x) = \begin{cases} +1 & w^T \phi(x) + b > 0 \\ -1 & \text{otherwise} \end{cases} \quad (2.8)$$

Given that there are many potential linearly-separating hyperplanes, SVMs choose the one that maximizes the margin of the training set. The margin of a single training sample x_i is the orthogonal distance to the hyperplane in $\mathbb{R}^j$. The margin of a training set is the smallest margin of all the training samples.

Intuitively, this margin is defined by the closest examples to the hyperplane. An SVM chooses a solution that maximizes the margin because such a solution

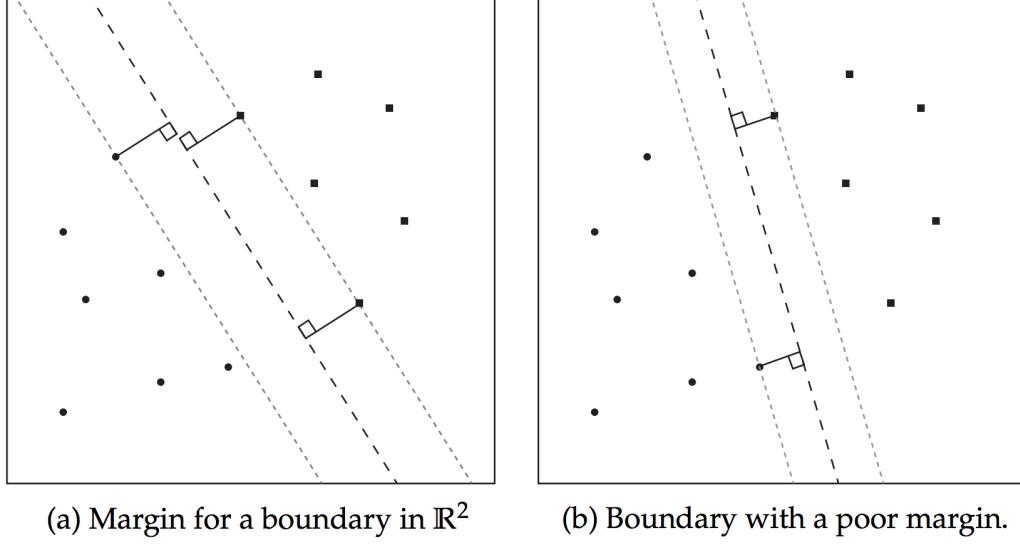


Figure 2.3: Examples of good and bad margins for SVMs. This figure first appeared in [Pfeffer et al. \(2014\)](#) as Figure 1.

would probably generalize better to other data compared to one with a smaller margin (Fig. 2.3). The problem of maximizing the margin can be reduced to the following optimization problem:

$$w^*, b^* = \arg \min_{w, b} \left\{ \frac{1}{2} \|w\|_2^2 \right\} \text{ s.t. } y_i \cdot (w^T \phi(x_i) + b) \geq 1, \forall i \in \{1, \dots, N\} \quad (2.9)$$

where N is the number of training samples and y_i is the target label for x_i .

The condition $y_i \cdot (w^T \phi(x_i) + b) \geq 1$ requires that the training data be linearly separable. However, in most cases, the data is not easily separable, even in higher-dimensional spaces. SVMs can be extended to have a “soft-margin” that allows some examples to be misclassified. Typically, this is done by introducing non-negative slack variables $\zeta_i \in \mathbb{R}$ for each training example x_i . Then, the new

problem can be formalized as follows:

$$w^*, b^* = \arg \min_{w, b} \left\{ \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^N \zeta_i \right\} \quad (2.10)$$

$$\text{s.t. } y_i \cdot (w^T \phi(x_i) + b) \geq 1 - \zeta_i, \forall i \in \{1, \dots, N\} \quad (2.11)$$

$$\zeta_i \geq 0, \forall i \in \{1, \dots, N\} \quad (2.12)$$

where $C > 0$. Perceptual annotations use this extension and replace the slack variables, which effectively induce a hinge loss function, with a human-weighted loss function ζ_ψ as formalized in 2.6. As previously mentioned, the regularizing parameter C determines how much simpler weights that maximize the margin are chosen over more complex ones that minimize the number of misclassified training samples.

One problem with the formulation given by Equation 2.9 is that it requires computing $\phi(x_i)$ for all $i \in \{1, \dots, N\}$. However, the feature space $\mathbb{R}^j$ that ϕ projects data points $x_i \in \mathbb{R}^d$ into can be very highly dimensional. This equation can be reformulated into a dual problem using Lagrange multipliers and kernel functions that is simpler to solve.

Using Lagrange multipliers $\alpha_1, \dots, \alpha_N \geq 0$, the dual problem is given by the following formulation:

$$\max_{\alpha \geq 0} \left\{ \sum_{i=1}^N -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \right\} \quad (2.13)$$

subject to $\sum_{i=1}^N \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$, where C is the regularizing parameter in Equation 2.10.

Because the feature space $\mathbb{R}^j$ is high-dimensional (and potentially even an infinite-dimensional space), computing $\phi(x_i)^T \phi(x_j)$ may be intractable. However, there exists special kernel functions that can compute the dot product of the higher-dimensional vectors by operating in the lower dimension vectors $x_i, x_j \in \mathbb{R}^d$.

Let's consider an example. Let $x \in \mathbb{R}^d$ and function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^j$ be defined

as follows:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_{d-1} \\ x_d \end{bmatrix}, \phi(x) = \begin{bmatrix} x_1^2 \\ x_1x_2 \\ x_1x_3 \\ \dots \\ x_1x_d \\ x_2x_1 \\ \dots \\ x_dx_{d-1} \\ x_d^2 \end{bmatrix} \quad (2.14)$$

Then, the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ can compute the dot product $\phi(x)^T \phi(z)$ as follows:

$$K(x, z) = (x^T z)^2 \quad (2.15)$$

$$= \left(\sum_{i=1}^d x_i z_i \right)^2 \quad (2.16)$$

$$= \left(\sum_{i=1}^d x_i z_i \right) \left(\sum_{i=1}^d x_i z_i \right) \quad (2.17)$$

$$= \sum_{i=1}^d \sum_{j=1}^d x_i x_j z_i z_j \quad (2.18)$$

$$= \sum_{i=1}^d \sum_{j=1}^d (x_i z_i) (x_j z_j) \quad (2.19)$$

$$= \phi(x)^T \phi(z) \quad (2.20)$$

Here, we see the advantage of a kernel function. Instead of needing to compute the entire feature map, ϕ , which is quadratic in this case, and then computing the scalar product directly, we only needed to compute the kernel function $K(x, z) = (x^T z)^2$, which is linear.

A popular kernel used in SVMs is the radial basis function (RBF) kernel, which is defined as follows:

$$K(x, z) = \exp\left(-\frac{1}{2\alpha^2} \|x - z\|_2^2\right) \quad (2.21)$$

Often, the function is simplified to the following form, where $\gamma = \frac{1}{2\alpha^2}$:

$$K(x, z) = \exp(-\gamma \|x - z\|_2^2) \quad (2.22)$$

One nice property of the RBF kernel is that its corresponding feature space has an infinite number of dimensions.

In conclusion, kernels can be used in SVMs to learn classifiers in some high-dimensional or even infinitely-dimensional space. For the complete derivations of the primal and dual formulations of SVMs, see [Bishop et al. \(2006\)](#).

2.3.1 Platt Probabilities

Oftentimes, one might not only be interested in a classification but also in the confidence of that prediction. Platt probability scores provide this for SVM classifications by approximating the posterior class probability $Pr[y = 1|x]$, that is, the probability that x is a positive example. [Platt \(2000\)](#) presents a method for estimating this posterior by the following sigmoid function:

$$Pr[y = 1|x] \approx Pr_{A,B}(f(x)) \equiv \frac{1}{1 + \exp(Af(x) + B)} \quad (2.23)$$

where f is a classification function and A and B are scalar parameters that are learned through a regularized maximum likelihood estimation from the training set. The problem to be solved is as follows:

$$A^*, B^* = \arg \min_{(A,B)} F(A, B) = - \sum_{i=1}^N (t_i \log(p_i) + (1 - t_i) \log(1 - p_i)) \quad (2.24)$$

$$\text{for } p_i = Pr_{A,B}(f(x_i)), \text{ and} \quad (2.25)$$

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = +1 \\ \frac{1}{N_- + 2} & \text{if } y_i = -1 \end{cases}, \forall i = \{1, \dots, N\} \quad (2.26)$$

where N_+ is the number of positive y_i 's and N_- the number of negatives samples in the dataset.

While [Platt \(2000\)](#) provides pseudocode for solving Equation 2.24, [Lin et al.](#)

(2007) propose an improved, more robust method that converges and implements it in the `libSVM` package that I use in the experiments described in Chapter 3.

2.4 Image Feature Representation

Oftentimes, performance in visual tasks related to semantic understanding is better when features that characterize various higher-level qualities of an image are used as training data instead of pixel values. For instance, object classification algorithms trained on direct pixel values often are disturbed by intra-class variance, such as the many orientations, colors, and sizes that a car can appear in an image. Strong features aim to capture the distinct characteristics of classes while ignoring the variance within them. In this section, I will succinctly describe two popular feature representations, histogram of oriented gradients (HOG) and convolutional neural networks (CNNs), that I use in this thesis.

2.4.1 Histogram of Oriented Gradients (HOG)

Dalal and Triggs (2005) introduced histogram of oriented gradients (HOG), a classic off-the-shelf algorithm for generating features that capture the local “slope-ness” of different parts of an image.

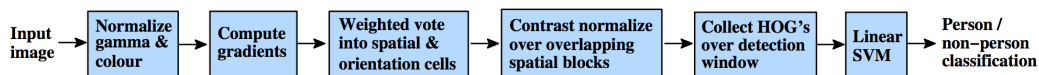


Figure 2.4: Histogram of oriented gradients (HOG) workflow. This figure appears in Dalal and Triggs (2005) as Figure 1.

The HOG algorithm contains the following key steps, which are included in Figure 2.4:

1. **Normalization.** Dalal and Triggs (2005) explains that normalizing the color distribution and gamma values in an image is optional as the next few steps also have a normalizing effect. Oftentimes, images are saved using gamma encoding, which records images in a way to save tones more

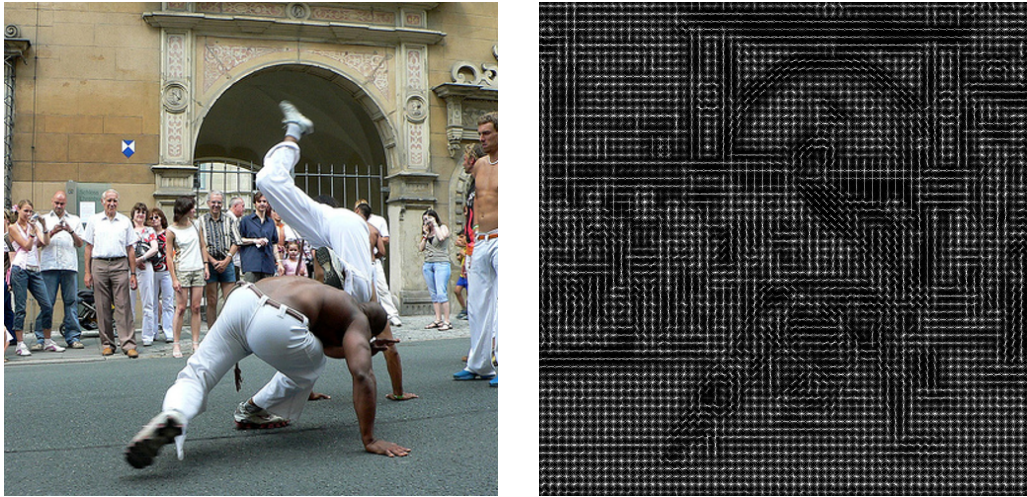


Figure 2.5: Example of HOG descriptor. The right image is a lossy, pictorial rendition of the HOG features generated for the left image using VLFeat’s `vl_hog` function and cell size of 8 (Vedaldi and Fulkerson, 2008).

efficiently. However, the technique is not meant to be used for displaying images. Gamma correction or normalization converts the light in a gamma-encoded image back to how it would have appeared in the original scene.

2. **Gradient Computation.** The horizontal and/or vertical gradient values of an image are computed by using a 1-dimensional derivative mask $[-1, 0, 1]$. For color images, gradient values are calculated for each channel separately and the one with largest norm is chosen for each pixel.
3. **Orientation Binning.** Then, the image is divided into sub-images called cells, the size of which is a parameter of the algorithm. For each cell, a histogram of edge orientations is created. For each pixel, a weighted vote for an edge orientation histogram channel based on its gradient vector is calculated; these votes are then accumulated and binned for a cell.
4. **Normalization Over Blocks.** After the previous step, an image can be described as a vector of the combined histogram entries for each of its cells. However, this representation is still sensitive to gradient variation across multiple cells. To counter this effect, cells are grouped into larger regions

called blocks that typically overlap. Within each block, its component cells' histograms are contrast normalized. The final HOG descriptor is then a vector of the components of the normalized cell histograms from the different blocks.

These descriptors can then be fed into a variety of computer vision algorithms, such as ones that detect and localize the presence of a person. HOG features describe local shapes well due to its use of gradients (Fig. 2.5). Additionally, by accumulating gradients into cells and normalizing them over blocks of cells, they are typically invariant to local geometric and photometric transformations. Lastly, cell and block sizes can be tuned for different applications. For instance, gesture detection might call for more fine-grained cell sizes while person detection may perform better with larger cell sizes.

2.4.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN), which are also known as ConvNets, are biologically-inspired artificial neural networks that are feed-forward, that is, they do not have cycles. CNNs mimic how the receptive field of neurons in the visual system communicate visual information to neurons with neighboring receptive fields and how this local information is pooled together and propagated through several layers of neurons. There exists a wide variety of CNN variants, yet most share common qualities in that they include a mix of convolutional and fully connected layers that perform various convolution, non-linear, and pooling operations that mimic the functions of biological neurons. By training a CNN on a large dataset like ImageNet and then feeding an image through the trained network, the output activation values can be used as image features. Libraries like Caffe (Jia et al., 2014) provide such trained CNNs for feature generation.

In this section, I will describe these main operations and the overall set-up and training of a CNN based on descriptions from (Bishop et al., 2006; DeepLearning, 2015; Rojas, 1996; UFLDL, 2015). For a more thorough treatment of neural networks, see Chapter 5 in Bishop et al. (2006).

The following operations are often performed in CNNs:

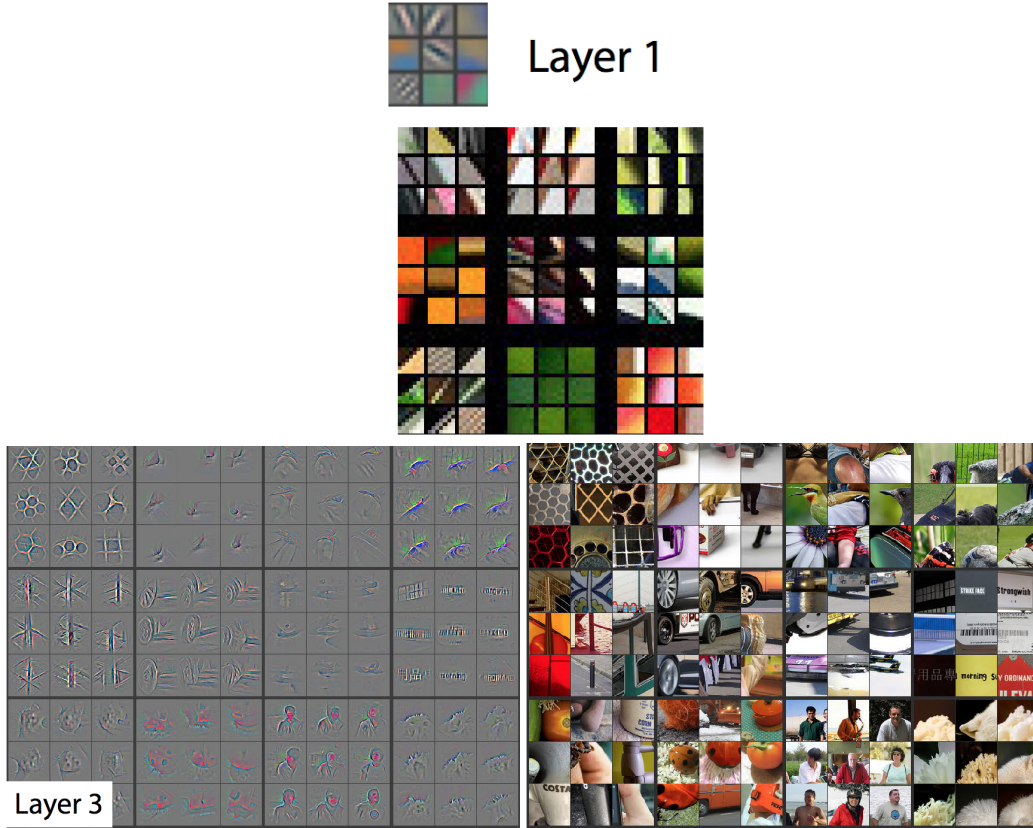


Figure 2.6: Visualization of neural network features. These layer visualizations of a fully trained CNN were first part of Figure 2 in [Zeiler and Fergus \(2014\)](#) and illustrate the different kinds of features captured by different layers in a CNN, with more low-level features like color being captured in layer 1 and higher-level features like parts of objects and creatures being captured in layer 3. The subimages shown in the rightside of each layer’s diagram corresponds to the 9 units in a random subset of a layer’s nodes with the largest activation values.

1. **Convolution.** The namesake of the algorithm, convolutional layers apply many 2-dimensional convolution kernels with weights learned via back-propagation to all the variables in a previous layer. For each kernel in this layer, the same weights are used throughout to constrain the number of parameters that need to be trained and to force a kernel to search for the same feature in all units in the previous layer. Such convolution operations extract out different features, with earlier convolution layers detecting low-level characteristics like edges and colors and later layers recognizing higher-level features (Fig. 2.6).
2. **Non-linearity.** Thresholding or activation functions like $f(x) = \tanh(x)$ and the sigmoid function $f(x) = \frac{1}{1+\exp(-x)}$ are often appended to a convolution step to add nonlinearity in CNNs, allowing for more complex features to be expressed.
3. **Pooling.** A pooling layer synthesizes information from several units in the preceding layer by calculating an average or maximum over them. This step allows local information captured in earlier layers to percolate to and influence later layers that capture higher-level features.
4. **Dropout.** Hinton et al. (2012) presents a simple yet novel way to prevent over-fitting, particularly in fully connected layers with lots of parameters, as well as to speed up training time by randomly dropping units throughout training.

CNNs consist of several layers, each of which contain units that mimic receptive fields in biological neurons (Fig. 2.7). Various operations are performed from layer to layer to generate complex features. Weights for kernels in the convolutional layers are learned via the backpropagation algorithm that iterates between two phases in order to calculate the partial derivatives needed to perform gradient descent. The propagation phase involves a forward and backward pass through the network. First, the feed-forward propagation phase simply calculates the values of each unit according to the operations of each layer and current weights of each convolutional layer. Then, for each output unit, the algorithm calculates the difference between its target and computed value and propagate the difference

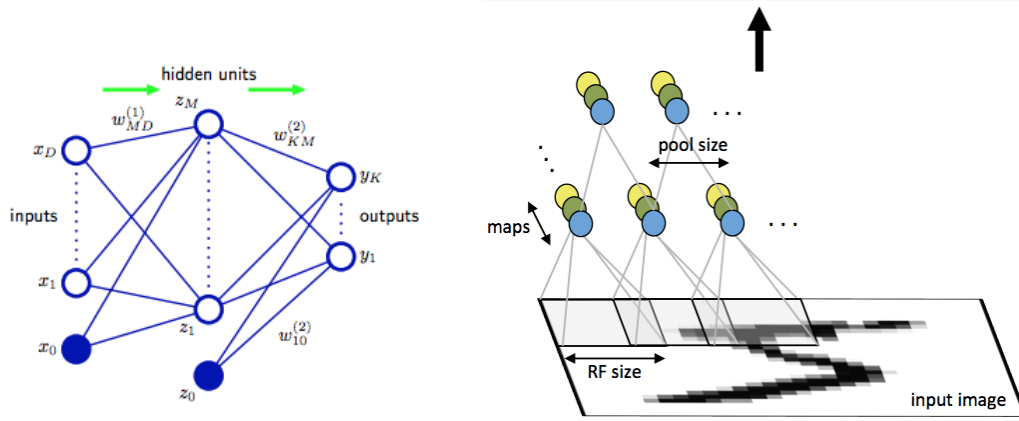


Figure 2.7: Network diagram. The left figure first appeared in [Bishop et al. \(2006\)](#) as Figure 5.1 and depicts a two-layer neural network with its input, hidden, and output layer as well as the weights of convolution kernels. The right figure first appeared in [UFLDL \(2015\)](#) as Figure 1 and shows pooling in the first layer of a CNN. It also illustrates how convolutional kernel weights are shared throughout the layer by coloring nodes that share the same weights.

backwards through the neural network to find the partial derivatives of weights at each layer. The update phase uses the partial derivative to correct for errors by subtracting a fraction of the weight's gradient from the weight. This fraction is a parameter called the learning rate that controls the tradeoff between how quickly the neural network is trained versus how finely trained it is. Backpropagation continues until some condition is satisfied, such as the gradients of weights being below a certain threshold.

Chapter 3

Experimental Design and Methods

The purpose of this chapter is to introduce the dataset and discuss the experiments and methods used in this thesis.

3.1 Data

3.1.1 Dataset

This thesis uses the dataset of images and corresponding fMRI recordings of subject S1 collected in [Stansbury et al. \(2013\)](#). The dataset provides 1386 500x500 color images of natural scenes, which subject S1 viewed while his brain activity was being recorded ([Stansbury et al., 2013](#)) used a training set of 1260 images and a test set of 126 images, thereby providing data for 1386 images).

The fMRI recordings were collected over six sessions. Each session consisted of several, five-minute runs that presented either a set of training images or a set of test images to the subject. A training run consisted of 36 training images presented two times each, while a testing run consisted of six images presented 12 times each. Images were included in exactly one run each and were randomly selected for each run, and there were 35 training runs and 21 test runs. In separate sessions, traditional functional localizers ([Spiridon et al., 2006](#)) were gathered and used to establish the locations of traditional regions of interest (ROIs) (see section

3. Experimental Design and Methods

2.1.1 for details on ROIs).

To constrain the dimensionality of the fMRI data, the time series recordings for each voxel were reduced to a single response amplitude. For each voxel, Stansbury et al. (2013) preprocessed the time series for each voxel as done in Kay et al. (2008) and then used these to estimate a hemodynamic response function. This function was then used to estimate a single-value response amplitude for each image. See Stansbury et al. (2013) for more details on data collection.

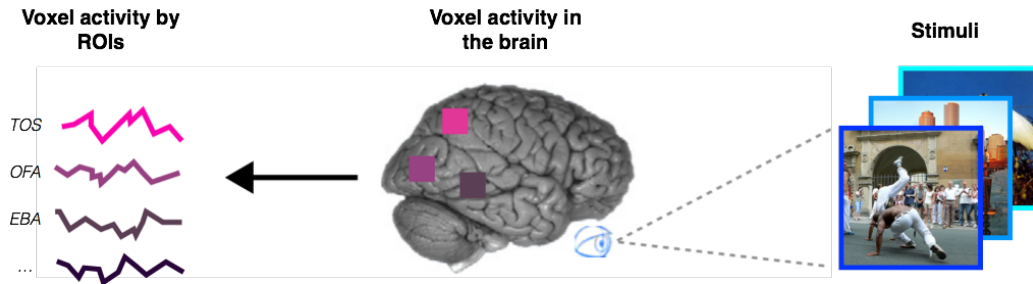


Figure 3.1: Recording brain activity. Stansbury et al. (2013) used fMRI to record BOLD voxel responses of humans viewing 1386 color images of natural scenes and labelled voxels in seven conventional functional ROIs: EBA, FFA, LO, OFA, PPA, RSC, and TOS.

3.1.2 fMRI Data Preprocessing

For each image, the Stansbury et al. (2013) dataset provides a response amplitude value for 67,600 voxels. The dataset has also labelled 3,569 of these voxels as being part of one of the following ROIs: V1, V2, V3, V3A, V3B, V4, LO, FFA, OFA, PPA, RSC, EBA, TOS.

Early VC	v1	v2	v3	v3a	v3b	v4	
Later VC	EBA	FFA	LO	OFA	PPA	RSC	TOS

Table 3.1: Regions of Interest (ROIs) for which I have labelled voxel activity, with the later visual cortical regions that I used for data in bold.

Because I am interested in semantic understanding of visual information, I focused on seven of these regions that are responsible for higher-level visual pro-

3. Experimental Design and Methods

cessing: EBA, FFA, LO, OFA, PPA, RSC, TOS (Tbl. 3.1, Fig. 3.1). 1,427 voxels belong to these regions.

In addition to the fMRI preprocessing described in Stansbury et al. (2013), I normalized the fMRI data. For each voxel, I calculated the minimum and maximum response amplitude, min and max , across all 1260 original training samples. Then, for each voxel, I rescaled all its response amplitude values across all 1386 samples using Equation 3.1, where x is a given voxel’s original response amplitude and x' is its scaled value.

$$x' = \frac{x - min}{max - min} \quad (3.1)$$

3.1.3 Image Pre-processing

I downsampled each of the original 500 x 500 colored images to be 250 x 250 grayscale images, with pixel values in the interval [0,1]. I also added a layer of Gaussian noise with a mean of 0 and variance of 0.01 to each image.

3.1.4 Label Assignment

The Stansbury et al. (2013) dataset provides per-pixel object labels for each image. I mapped 271 of these labels to one of five object categories: HUMANS, ANIMALS, BUILDINGS, FOODS, and VEHICLES (see Appendix A for the mappings).

For each image and for each of the five object categories, if $> 20\%$ of an image’s original pixels’ labels were part of a given object category, I tentatively labelled that image as a positive sample for that category. If an image was labelled a positive sample for more than one object category, I labelled it as a negative sample for all categories.

170 images were initially labelled as positive samples for multiple categories. After they were removed, 646 images that each were labelled with a single object category remained. There were 219 HUMANS images, 180 ANIMALS images, 151 BUILDINGS images, 59 FOODS images, and 37 VEHICLES images.

3. Experimental Design and Methods

	humans	animals	buildings	foods	vehicles
# of Positive Sample	219	180	151	59	37

Table 3.2: Number of Images in Each Object Category.

3.1.5 Image Features

For each image, I generated histogram of oriented gradients (HOG) descriptors with a cell size of 32 using the VLFeat library’s `vl_hog` function (Vedaldi and Fulkerson, 2008), which computes UoCTTI HOG features (Fig. 3.2).

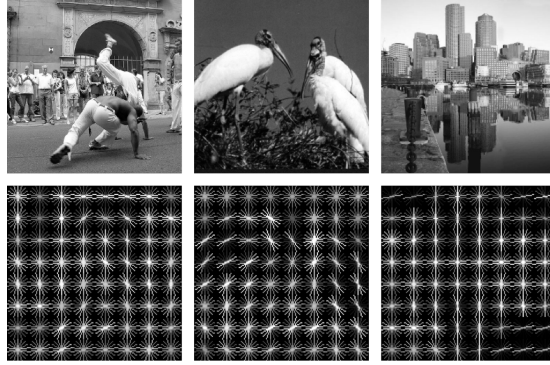


Figure 3.2: HOG features were generated with VLFeat library’s `vl_hog` function using a cell size of 32.

For each image, I generated convolutional neural networks (CNN) features using the Caffe library’s BLVC Reference CaffeNet model (Jia et al., 2014), which is AlexNet trained on ILSVRC 2012, with a minor variation from the version as described in Krizhevsky et al. (2012) (Fig. 3.3).

See preliminary sections 2.4.1 and 2.4.2 for background on HOG and CNN features.

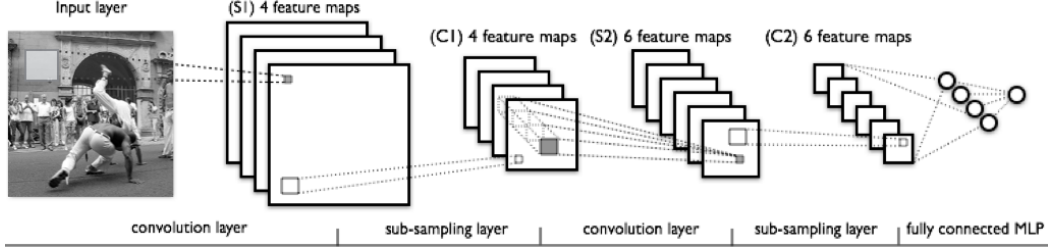


Figure 3.3: CNN features were generated using the Caffe deep learning framework (figure modified from [Krizhevsky et al. \(2012\)](#)).

3.2 Experimental Design

3.2.1 Data Partitions

I created five training and test partitions, in which 80% of the data was randomly designated as training data and the remaining 20% was designated as test data.

3.2.2 Experiments

For each random data partition, I ran experiments for the 127 ways that the 7 higher-level visual cortical regions could be combined (Eqn. 3.2; see [Appendix B](#) for a list of the 127 combinations and the number of voxels in each combination).

$$\binom{7}{1} + \binom{7}{2} + \dots + \binom{7}{7} = 127 \text{ combinations} \quad (3.2)$$

In each experiment, for a given combination of higher-level visual cortical regions and for a given object category, I did the following:

1. Generate perceptual annotations from Platt probabilities of an RBF-kernel SVM classifier trained on the training voxel data for that combination.
2. Create five balanced classification problems. For each balanced classification problem and each set of image descriptors (HOG and CNN features), train and test two SVM classifiers —one that uses a standard hinge loss (HL) function and another that uses a human-weighted loss (HWL) function —using an RBF-kernel (see section 2.2.2 and 2.3 for more details about

3. Experimental Design and Methods

loss functions and kernel functions respectively).

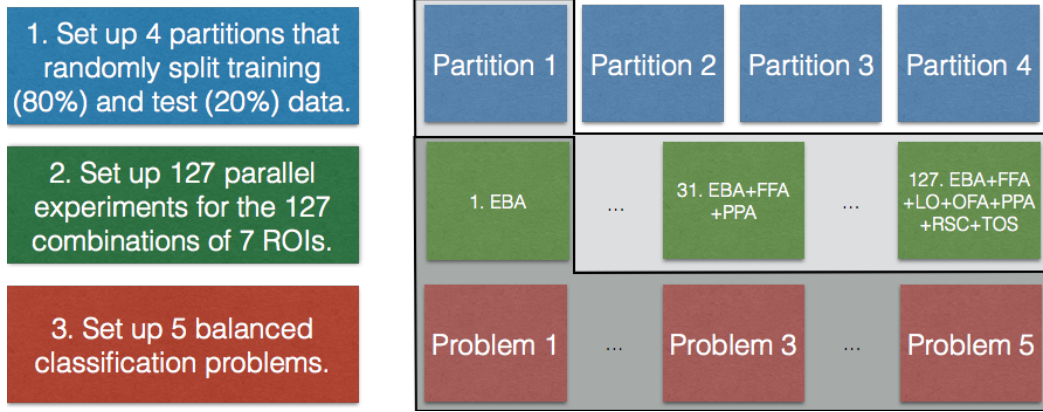


Figure 3.4: Experiment Design.

I used the libSVM package to train and test SVM classifiers using a hinge loss function (Chang and Lin, 2011). To train a model using a human-weighted loss function, I used code made publicly available from Scheirer et al. (2014) and applied a patch to libSVM to support an SVM-variant that uses a human-weighted loss function.

3.2.3 Generating Perceptual Annotations from Brain Activity

The main challenge of generating perceptual annotations from brain activity lies in reducing highly-dimensional, nonlinear data to a salient, lower-dimensional signal of “learnability”. The perceptual annotation algorithm described in Scheirer et al. (2014) requires a single value as a perceptual annotation. I tried a variety of traditional metrics to capture the distance of a given sample from the “average” voxel activity for a positive HUMANS image, such as applying a PCA transform to voxel activity and examining distances in the first few principal components. Yet, these basic approaches failed to produce interpretable annotations. In this section, I present a novel procedure for Platt probability perceptual annotations (Platt, 2000) generated from SVM classifiers (SVCs) trained on brain activity.

3. Experimental Design and Methods

For each object category and for all voxels from a given combination of ROIs, I first identified all the positive samples for that object category as well as all the samples that are negative for all object categories. I only generated annotations for this subset of a partition’s training set. This constraint maximizes the signal-to-noise ratio in the annotations and improves the saliency of annotations for a specific object category by only annotating clear positive and negative samples.

Then, I randomly split the voxel data for each of the images in this subset into five folds. For each of these folds, I held out the current fold as test data and combined the other four folds as training data. With this newly formed training set, I conducted a parameter search for the best c and γ parameters to be used in a RBF-kernel, binary SVC for the current object category. To do this, I iterated through all combinations of $c, \gamma \in \{2^{-7}, 2^{-6}, \dots, 2^6, 2^7\}$ and chose the best c and γ that yielded the highest accuracy on a five-fold cross-validated SVC. Then, I trained an SVC using the best c and γ parameters on all the training data from the four folds. Finally, I tested the classifier on the held-out fold and generated Platt probability scores for each sample in the held-out fold.

Figure 3.5 illustrates how a classifier trained on other voxel data is used to generate Platt probability perceptual annotations for the HUMANS category on a held-out set of data. This diagram shows how Platt probabilities are related to the distance from the decision boundary. Furthermore, Figure 3.5b, which shows actual generated Platt probabilities generated by my algorithm, qualitatively demonstrates the saliency of this procedure in reducing high-dimensional brain activity to a single value that captures the “learnability” of an image. The extreme Platt probabilities scores of images 1, 2, 7, and 8 can be explained by the scenes they depict: Images 1 and 2 clearly depict scenes of man-made structures such as buildings, roads, and cars, while images 7 and 8 depict social scenes involving multiple humans. Similarly, the scores of images 3-6, which are either of animals or both humans and critters, are consistent with the body of neuroimaging literature that suggests that similar areas in the visual cortex are activated by humans and animals (Çukur et al., 2013; Gauthier et al., 2000b; Kanwisher et al., 1997).

3. Experimental Design and Methods

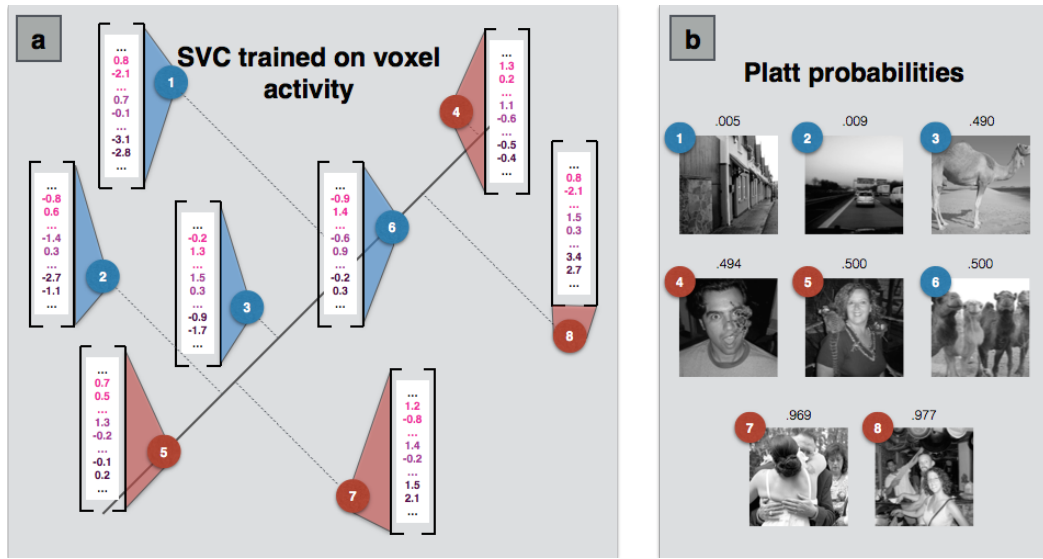


Figure 3.5: Generating Platt probability perceptual annotations from brain activity. This example shows how annotations for the HUMANS category are generated for a held-out set of images using a RBF-kernel SVC trained on the voxel activity of another set of images. The right panel presents actual Platt probability scores generated by the algorithm outlined in 3.2.3. Positive HUMANS samples are denoted by red circles, while negative ones are denoted by blue circles.

3.2.4 Balanced Classification Problem

For each object category, combination of higher visual cortical regions, and set of image descriptors, I set up five balanced classification problems. For each problem, I first found the minimum of the number of positive samples and the number of negative samples in the training set. Then, I randomly chose that number of positive examples and that number of negative examples to be used in a balanced classification problem. (For all object categories, because there were always more negative than positive samples, all positive samples were used in the balanced problem and the same number of negative samples were randomly selected.) The balanced problems only balanced the training data; each problem used the same test set: the partition’s held-out test set.

Then, I trained two RBF-kernel SVCs on image features, with $c = 1$ and $\gamma = 1/\text{num_feats}$. One classifier uses a standard hinge loss function while the other classifier uses a human-weighted loss function (see section 2.2.2 for details on loss functions). The latter classifier that uses a human-weighted loss function incorporates Platt probability perceptual annotations as additive weights (see section 2.2). For samples without annotations, additive weights of 0.0 are used. Finally, for each trained model, I tested the classifier on the partition’s test set.

Figure 3.6 illustrates how using perceptual annotations in an SVC that uses a HWL function changes the boundary. In this toy diagram, the new boundary in Fig. 3.6b is farther away from strong positive images as well as weak negative images. The figure also demonstrates how not all images need perceptual annotations by including the two images that are displayed here without a shadow and thick border.

To prevent overfitting, I adapted the concept of dropout described in [Hinton et al. \(2012\)](#), which demonstrated that randomly dropping units from a neural net during training greatly improves the model’s ability to generalize to other data compared to other regularization techniques. Recall that I only generated annotations for the positive samples for a given object category and the samples that are not part of any of the five object categories; this subset was around 45% of each partition’s training set. Then, when creating the balanced problems described in this section, the balanced training set is randomly selected without

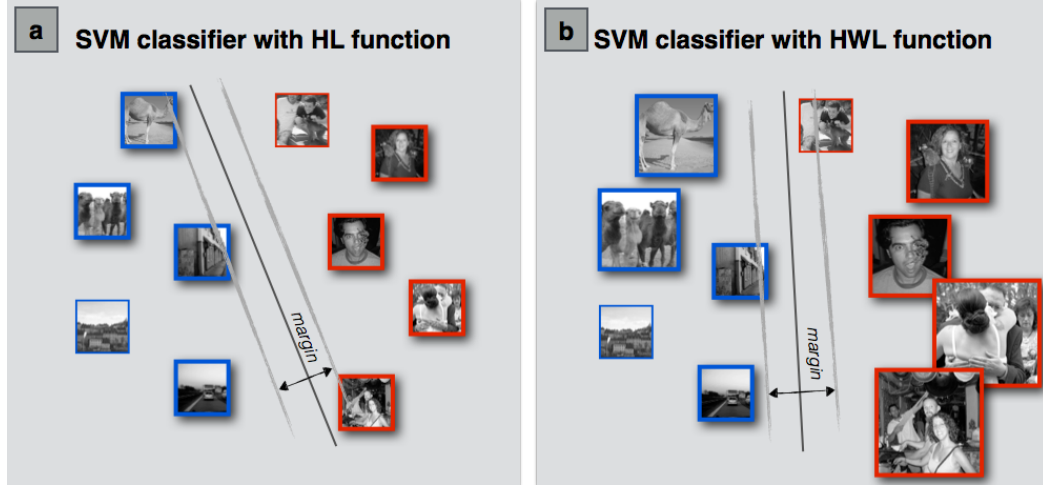


Figure 3.6: Re-weighting training samples using perceptual annotations. This figure show how an SVC using a HWL function chooses a decision boundary that is farther away from images with larger perceptual annotations. The sizes of the images in this figure are weighted by the generated Platt probability perceptual annotations in Fig. 3.5.

considering which samples have annotations, a regularization step inspired by the dropout technique.

In the next few chapters, I will present results that support this thesis’s primary argument: the use of perceptual annotations as described in this chapter selects solutions that are more like the ones humans would choose and are ultimately better generalized models that move us closer to closing the gap between human and machine performance in semantic understanding tasks.

Chapter 4

Results and Analysis

In this chapter, I will explain my experimental results and analysis. First, I will revisit my experimental set-up and introduce the basic statistics I computed on my results. Then, I will demonstrate the significant improvements my algorithm had on object classification tasks as well as compare how using different image features impacted the observed gains. Next, I will examine how different combinations of ROIs yielded varying results and analyze the influence of individual ROIs in improving classification performance. Finally, I will compare how using different perceptual annotation weighing schemes improve classification performance.

4.1 Experimental Set-up

4.1.1 Experiments

As previously described, I ran my experiments using four partitions of randomly split training and test sets. Because there are 1386 total datum, each partition has 1109 datum in the training set and 277 datum in the test set to yield a 80% – 20% training – test split. For each partition, I setup five balanced classification problems. For each balanced classification problem, I recorded the number of test datum correctly classified by RBF-kernel SVCs.

Thus, I have a total of 20 balanced experiments for each permutation of experimental conditions, that is, for each of the two image feature sets (i.e. HOG

features, CNN features), two SVM loss functions (i.e. hinge loss (HL) function, human-weighted loss (HWL) function), five object categories (i.e. HUMANS, ANIMALS, BUILDINGS, FOODS, VEHICLES), and 127 combination of ROIs (i.e. EBA, ..., EBA+FFA+PPA, ..., EBA+FFA+LO +OFA+PPA+RSC+TOS).

4.1.2 Simplifying Assumption

In the subsequent analysis, I treated the results from these experiments as i.i.d., despite the latent, intrinsic dependencies between them.

4.1.3 Note about Vehicles Category

In this chapter, I will focus on results for the following four object categories: HUMANS, ANIMALS, BUILDINGS, FOODS. Brain activity annotations provided no significant improvements in VEHICLES classification. This was most likely due to a sparsity of positive VEHICLES training datum; there are only 37 VEHICLES images in the whole dataset (Tbl. 3.2).

4.2 Basic Statistics

For each set of image features, object category, SVM loss function, and combination of ROIs, I computed the sample mean, standard deviation, and standard error of the number of correctly classified test datum.

Let $n = 20$ denote the number of experiments, $t = 277$ denote the number of test datum per partition, and $z_1, \dots, z_n \in [0, t]$ denote the number of correctly classified test datum. Then, the sample mean (Eqn. 4.1), standard deviation (Eqn. 4.2), and standard error (Eqn. 4.3) of the number of correctly classified test datum are given by the following equations:

$$\mu = \frac{\sum_{i=1}^n z_i}{n} \quad (4.1)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (z_i - \mu)^2}{n - 1}} \quad (4.2)$$

$$s.e. = \frac{\sigma}{\sqrt{n}} \quad (4.3)$$

Unless otherwise specified, results bar graphs plot the mean percentage of correct classification and the standard error of the 20 experiments under certain experimental conditions.

4.2.1 Significance Testing

Unless otherwise specified, one-tailed, paired t-tests were used to test whether classification performance when using perceptual annotations (i.e. using a HWL function) was significantly better than when not using any perceptual annotations (i.e. using a HL function). Double asterisks (**) denotes 99% confidence interval while a single asterisk (*) denotes 95% confidence interval.

4.3 Basic Results

Figures in this section compare the average performance on held-out test data of 20 experiments that use RBF-kernel SVM classifiers (SVCs) with a hinge loss (HL) function and another 20 experiments RBF-kernel SVCs with a human-weighted loss (HWL) function to incorporate perceptual annotations based on brain activity. Via paired, one-tailed significance testing, Figure 4.1a shows that using perceptual annotations trained on all higher visual cortical regions significantly improves classification accuracy across all four object categories.

A significant amount of fMRI decoding literature focuses on three ROIs: EBA, FFA, and PPA (Cox and Savoy, 2003; Kriegeskorte et al., 2007; Spiridon et al., 2006). This is in part because these three regions are thought to respond to visual cues of high research interest: body parts, faces, and places respectively. Given the overlap between our object categories and these visual cues, I hypothesized that perceptual annotations generated from brain activity in these three regions would significantly improve classification accuracy. A comparison of SVCs that use perceptual annotations based on brain activity from these three regions and SVCs that use no annotations aligns with neuroimaging literature (Fig. 4.1b). These results suggest that brain activity from the EBA, FFA, and PPA regions

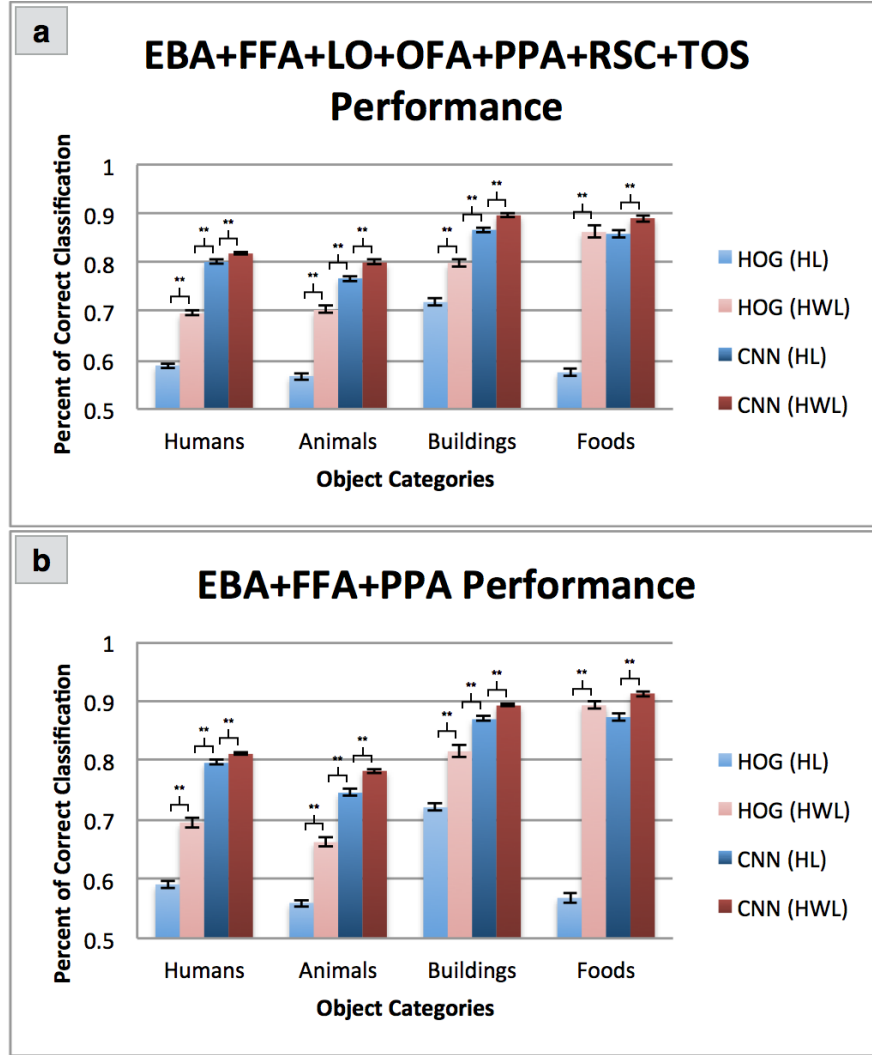


Figure 4.1: Side-by-side comparisons of the mean classification accuracy between RBF-kernel SVCs —using both HL and HWL functions —that use HOG features and ones that use CNN features. The top graph shows results of experiments that generated perceptual annotations from all seven higher-level visual ROIs, while the bottom graph shows results of experiments that generated annotations from EBA, FFA, and PPA. Paired, one-tailed t-testing was used for significance testing between adjacent representations of the four combinations of image features and SVC loss function.

significantly inform and improve the classification of my object categories of interest.

Not only do we observe significant improvements in classification accuracy when perceptual annotations were generated from voxels in all 7 ROIs or from voxels in the EBA, FFA, and PPA regions, we also observe that using perceptual annotations significantly increased classification accuracy when perceptual annotations were generated from most of the 127 ROI combinations of voxels (Fig. 4.2).

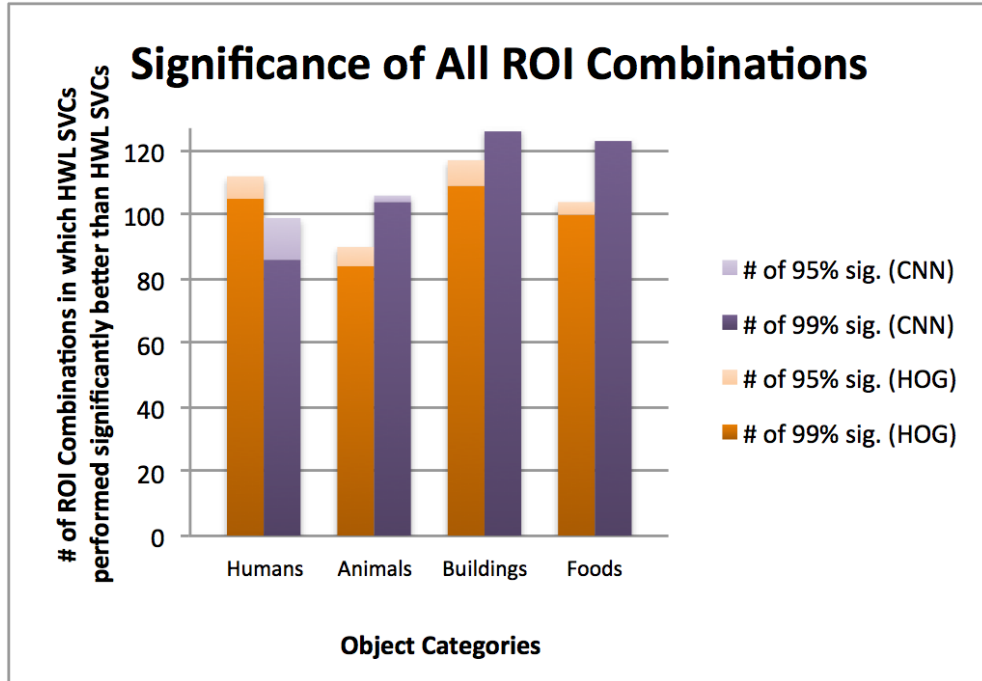


Figure 4.2: Significance of all ROI combinations. The number of combinations of ROIs (out of 127 total combinations) in which 20 SVCs that incorporated perceptual annotations by using a HWL function performed significantly better than 20 SVCs that used a standard HL function are plotted here. 1-tailed, paired t-tests were used to test significance.

4.3.1 Comparison of HOG vs. CNN Image Features

Comparing the classification accuracy between SVCs using HL functions that use HOG features and SVCs that use CNN features, we observe that the performance gap is typically more than halved when SVCs using HWL functions are trained on HOG features (Fig. 4.1). These results suggest that improvements in decision making (i.e. the use of fMRI annotations) may be able to compensate for poor feature representation.

Additionally, adding perceptual annotations based on brain activity provides a larger improvement to classifiers using HOG features compared to those using CNN features (4.1). These results suggest that some of the information that brain activity annotations contain may already be latently captured in the CNN features. I generated these CNN features using BLVC’s Caffe model, which was trained on the large ImageNet dataset (Russakovsky et al., 2014). Because the neural net model was trained on such a large and diverse dataset, the CNN features generated by this model probably captured latent information about how similar (or dissimilar) a given image was to many different kinds of objects contained in the ImageNet dataset. Thus, the smaller improvement to classifiers using CNN features can be attributed to not only to the fact that CNN features are better features, but also because they include latent information similar to that captured by brain activity annotations.

4.4 Results by Combinations of ROIs

4.4.1 Feature-Level Fusion of ROIs

I demonstrated that perceptual annotations based on fMRI activity from the EBA, FFA, and PPA regions significantly improve classification accuracy (Fig. 4.1b). Yet, when we compare the classification accuracy between SVCs that used no annotations and ones that used annotations based on fMRI activity from only the EBA region, only the FFA region, and only the PPA region, we no longer observe consistent, significant improvements in the classifiers that used annotations based on activity from only the FFA region and only the PPA region (Fig. 4.3). The difference in performance between SVCs that used annotations

based on activity from individual ROIs and ones that used annotations based on activity from the combination of the three ROIs can be explained by the feature-level fusion of information from the three ROIs.

Information fusion refers to the practice of merging multiple sources of information (often from different modalities, such as fMRI and EEG technology) into a single, more comprehensive source of data (Mahler, 2007). The resulting, new dataset often contains latent, synergetic relationships between the original, individual data sources. Experiments that combine brain activity from EBA, FFA, and PPA exhibit a similar phenomenon by fusing feature-level information together from the different areas.

The improved performance when multiple ROIs are used to generate annotations suggests that the EBA, FFA, and PPA regions synergistically work together in object detection tasks (Figs. 4.3 and 4.1b). When adding additional regions one by one to EBA in order to generate annotations (first FFA, then PPA), we see significant gains in the FOODS category (4.4). Neuroimaging research suggests that specific ROIs are activated by stimuli from the HUMANS and BUILDINGS categories (Epstein and Kanwisher, 1998; Gauthier et al., 2000b; Kanwisher et al., 1997), yet there is no FOODS brain region. Thus, this result suggests that although there is no single FOODS brain area, several, possibly disparate patches of voxels across the EBA, FFA, and PPA regions may work together in detecting FOODS.

4.4.2 Top Combinations of ROIs

After observing impressive results from experiments that used brain activity from all available higher-level visual cortical regions and from the 3 well-researched object recognition ROIs (i.e. EBA, FFA, and PPA), I examined which of the 127 combinations of ROIs yielded the best classification accuracy rates (Figs. 4.5, 4.6, 4.7, 4.8).

When looking at the ROIs included in the top five combinations for each object category and across both HOG and CNN image features (Tbl. 4.1), we see some unsurprising results: EBA, which selects for the human body and body parts (Downing et al., 2001), is included in all top ROI combinations for the

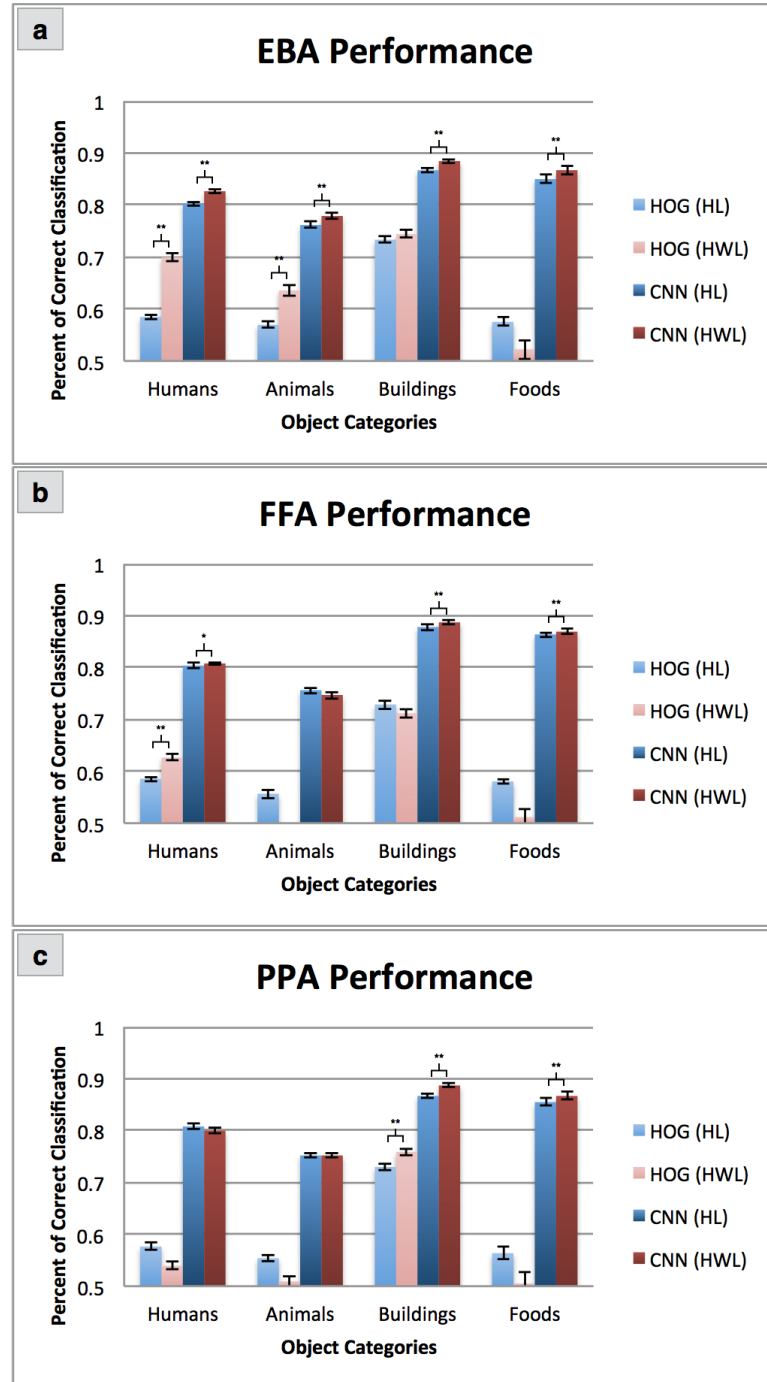


Figure 4.3: Results from Individual ROIs: EBA, FFA, and PPA.

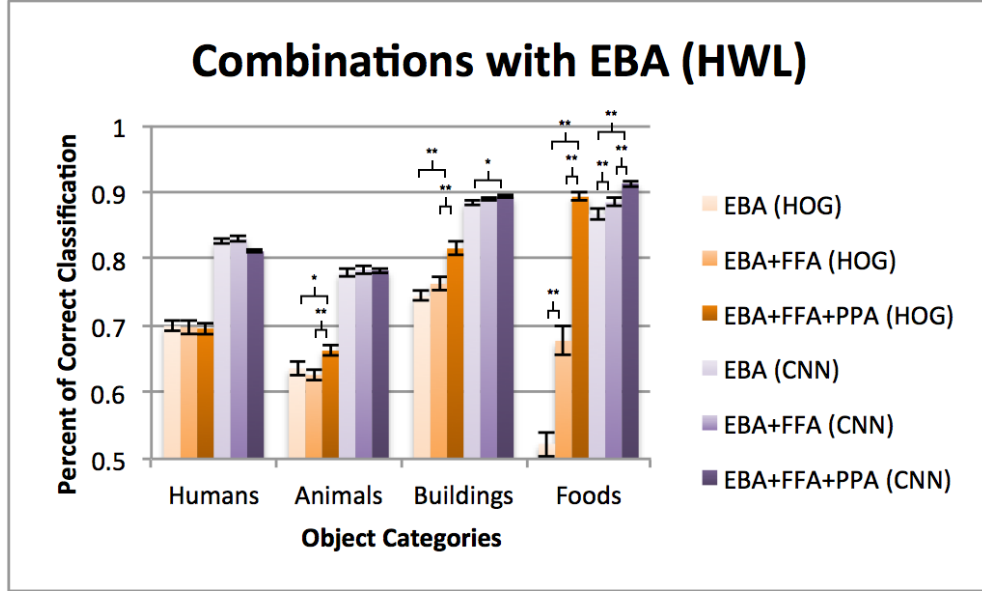


Figure 4.4: Comparison of ROI Combinations that include EBA.

HUMANS and ANIMALS categories. Similarly, the OFA region, which assists in the recognition of human faces (Gauthier et al., 2000b), is highly represented in the top ROI combinations for the HUMANS and ANIMALS categories. Also as expected, nearly all top ROI combinations for the BUILDINGS category include the PPA region, which is activated by “places” (Epstein and Kanwisher, 1998).

However, we also see some surprising outcomes: All top ROI combinations for the FOODS category include the EBA, PPA, and the FFA regions. The last area is selective for human faces (Kanwisher et al., 1997), and none of these regions is known to be selective for foods. Additionally, activity in EBA and FFA —areas associated to human body and face detection —appeared relatively influential in the classification of BUILDINGS. This preliminary exploration encouraged me to analyze the impact of including individual ROIs using more formal and rigorous statistical methods.

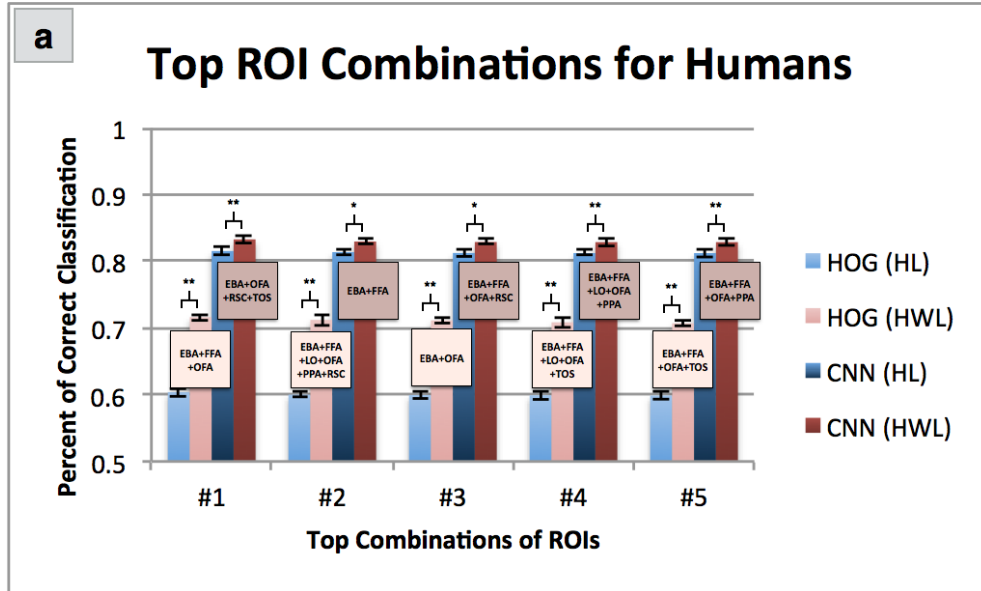


Figure 4.5: Top five combinations of ROIs for HUMANS.

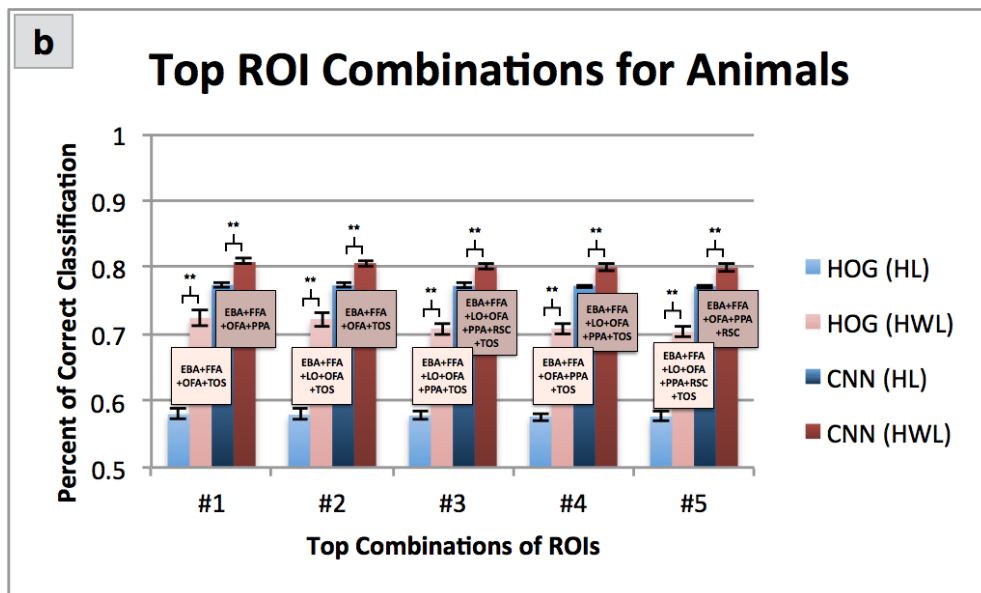


Figure 4.6: Top five combinations of ROIs for ANIMALS.

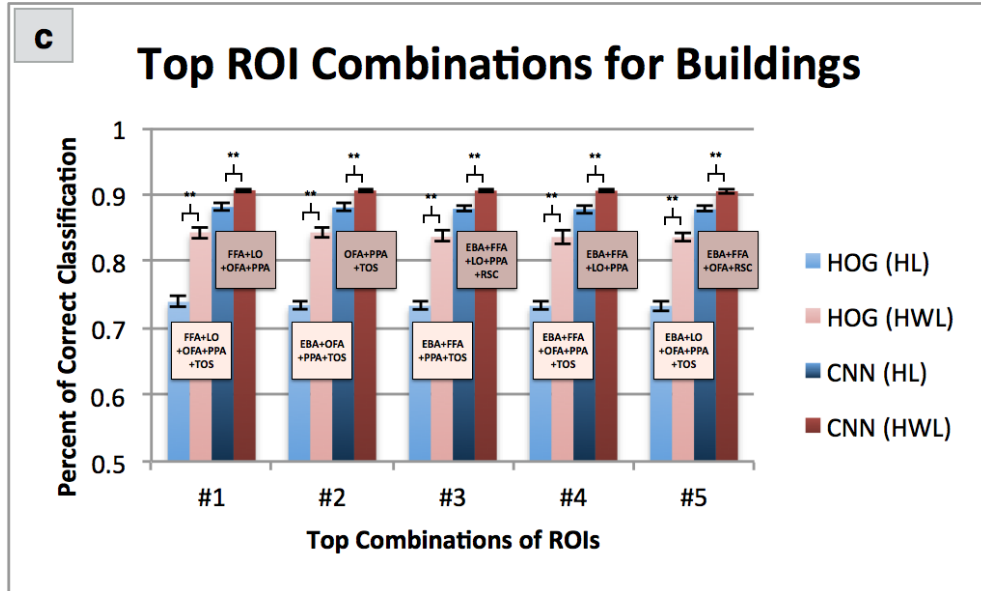


Figure 4.7: Top five combinations of ROIs for BUILDINGS.

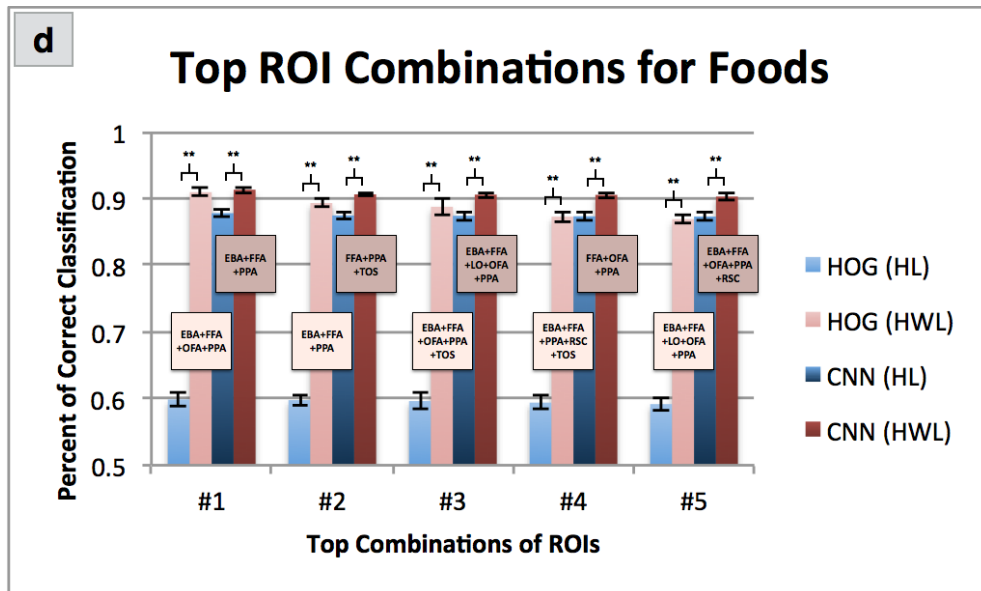


Figure 4.8: Top five combinations of ROIs for FOODS.

	EBA	FFA	LO	OFA	PPA	RSC	TOS
humans	10	8	3	9	3	3	3
animals	10	10	5	10	7	3	8
buildings	8	7	5	7	9	2	6
foods	10	10	2	6	10	2	3

Table 4.1: Count of ROIs included in the top five combinations for each object category. The counts from experiments using HOG features and those using CNN features were combined (Figs. 4.5, 4.6, 4.7, 4.8), leading to a maximum count of 10.

4.5 Analysis of ROIs

First, to begin analyzing the influence of individual ROIs, I expanded from looking at the top five combinations for each object category to looking at all 127 combinations in experiments that leverage fMRI-based annotations by using HWL. Each ROI (i.e. EBA) was included in 64 of the 127 combinations. Thus, for each ROI, I counted how many of the 64 combinations that included a given ROI (i.e. EBA) had a mean classification accuracy greater than the average accuracy over all 127 combinations for experiments that use HWL for a given set of image features (Fig. 4.9).

As an example, I will explain how I calculated the EBA statistic in Fig. 4.9a. Consider all the perceptually annotated experiments, that is, ones that incorporate annotations using HWL, that use HOG features to classify HUMANS. The mean classification accuracy over all 127 sets such experiments is 65.09%. Then, to calculate the EBA statistic shown in Fig. 4.9a, I examined the 64 sets of results under the same experimental conditions (i.e. HOG features, HWL function) that used voxels from EBA to generate perceptual annotations. For each set of experiments corresponding to a combination of ROIs that includes EBA, I checked whether its mean classification accuracy was greater than 65.09%. Out of all 64 sets of experiments that used EBA to generate annotations, 96.88% of them had an above-average mean classification accuracy, i.e., one that exceeded 65.09%.

The cursory analysis captured in Fig. 4.9 appears to confirm the importance of EBA for classifying HUMANS and ANIMALS, as 96.9% and 95.3% of combinations

that included EBA in HOG experiments had above-average classification accuracy rates in HUMANS and ANIMALS object classification tasks (Fig. 4.9a). Likewise, it also affirmed the influence of PPA for classifying BUILDINGS and FOODS.

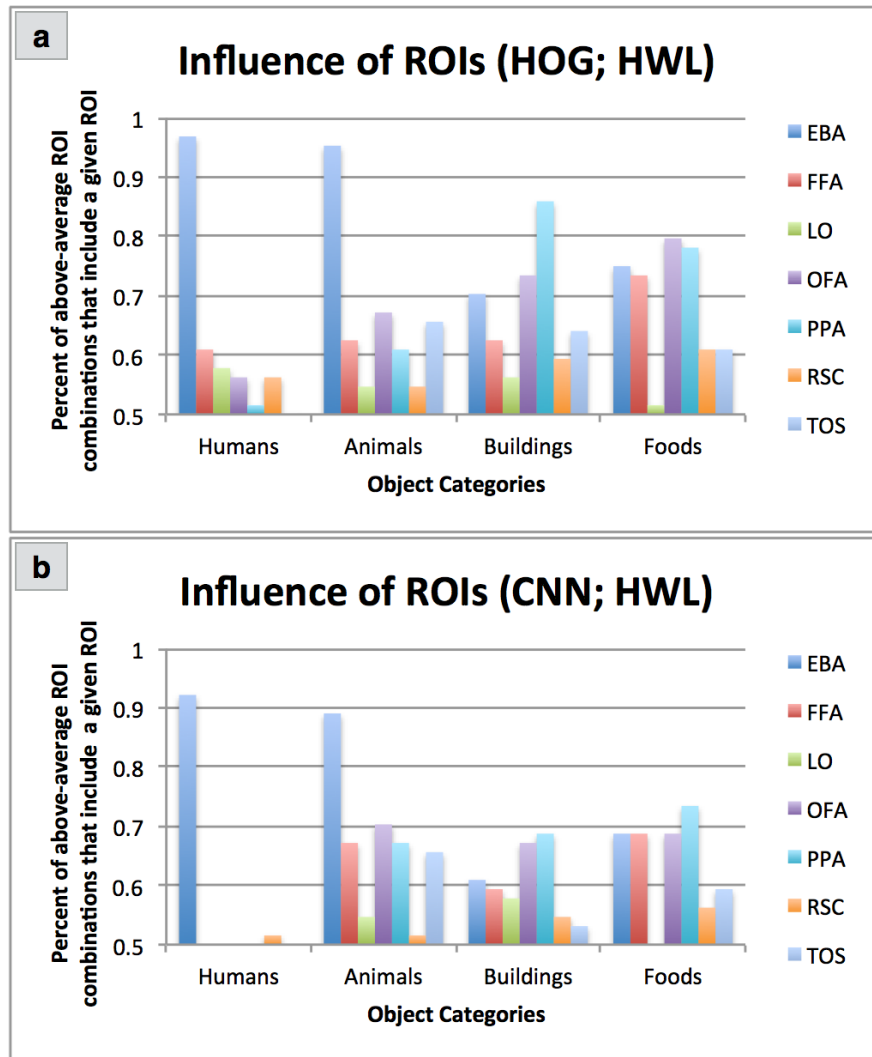


Figure 4.9: Influence of individual ROIs as perceptual annotations in classifiers using a HWL function. I counted how often the 64 ROI combinations containing a specific ROI (i.e. EBA) had a mean classification accuracy greater than the mean accuracy over all 127 sets of experiments.

4.5.1 Null Distributions

I was interested in testing whether the above-average accuracy rates observed in HUMANS and ANIMALS classification experiments that included EBA as well as in BUILDINGS and FOODS experiments that included PPA was statistically significant or a product of random chance. Thus, I generated null distributions with which to compare the above-average accuracy of experiments containing an individual ROI (i.e. EBA).

For each object category and set of image features, I generated a null distribution with 1,000,000 samples. Each sample in the null distribution captures how often a random set of 64 combinations would might reasonably have an above-average classification accuracy. Such a null distribution is appropriate because we are comparing how often a specific set of 64 combinations that all include a given ROI has an above-average classification accuracy. We are testing the significance that individual ROIs have on generating salient perceptual annotations that yield above-average classification accuracy rates. Thus, these samples simulate randomly assigning ROI labels to the 127 combinations. If individual ROIs did not significantly contribute to the above-average accuracy rates observed, we should observe that the above-average accuracy rates of combinations that include specific ROIs should fall near the mean of the null distribution.

To generate each of the 1,000,000 samples, I randomly selected 64 ROI combinations. Then, I counted how many of those 64 randomly selected combinations have a mean classification accuracy that is greater than that of all 127 sets of experiments corresponding to the 127 total ROI combinations. I saved this count as a fraction out of 64 to serve as the new sample.

Consider how a sample for the HUMANS, HOG null distribution is generated. As previously mentioned, the mean classification accuracy over all 127 sets of perceptually-annotated HUMANS experiments that use HOG features is 65.09%. I randomly chose 64 of those 127 sets of experiments. For each one of those 64 sets of experiments, I checked whether or not it has a mean classification accuracy greater than 65.09%. I counted the number of sets that did have an above-average accuracy and saved this number as a fraction out of 64. Suppose 35 out of the 64 randomly selected experiment sets had a mean classification accuracy that was

above 65.09%. Then, the generated sample would be $\frac{35}{64} = 0.5469$.

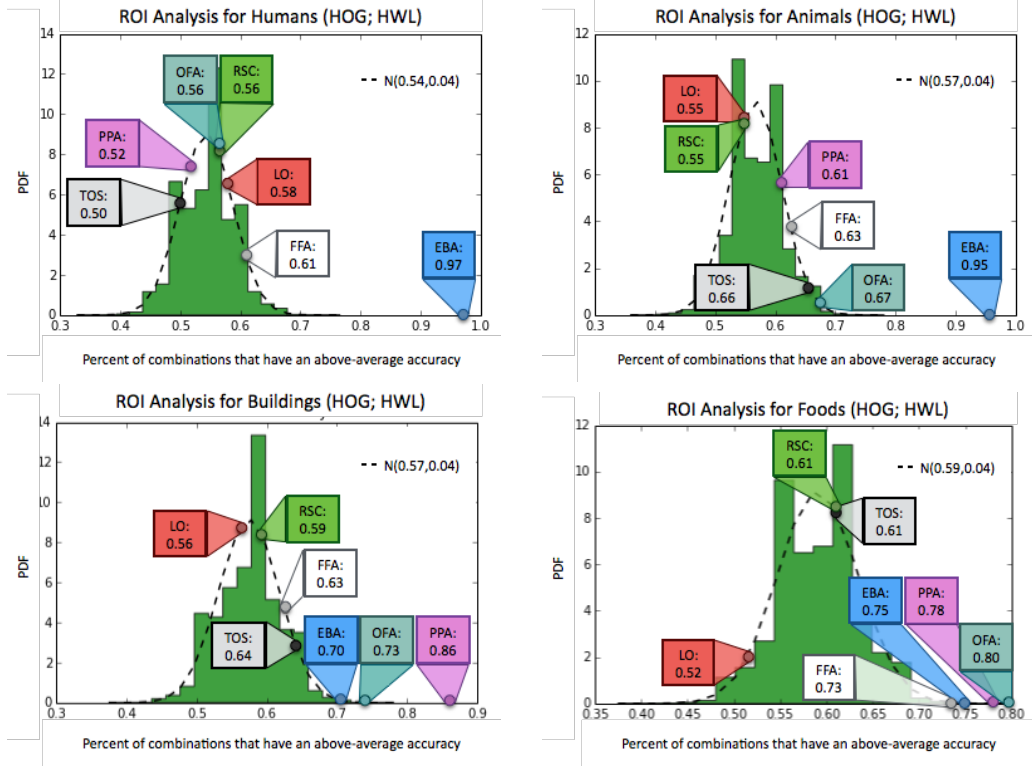


Figure 4.10: Null distributions for analysis of ROIs (HOG). For each ROI, the percent of combinations that include it and that have an above-average classification accuracy (out of all 64 combinations that include a given ROI) is plotted and labelled; these percents are also graphed in Fig. 4.9. Histograms of the null distributions described in section 4.5.1 are plotted as well as Gaussian curves fitted to them.

Figures 4.10 and 4.11 plot the generated null distributions as histograms for each set of image features and object category. I also fit a Gaussian curve to each of these null distributions using the `scipy.stats.norm` package. Lastly, Figures 4.10 and 4.11 include the actual percentages for each ROI from Figure 4.9 of how many of the 64 ROI combinations containing that region have an above-average mean classification accuracy.

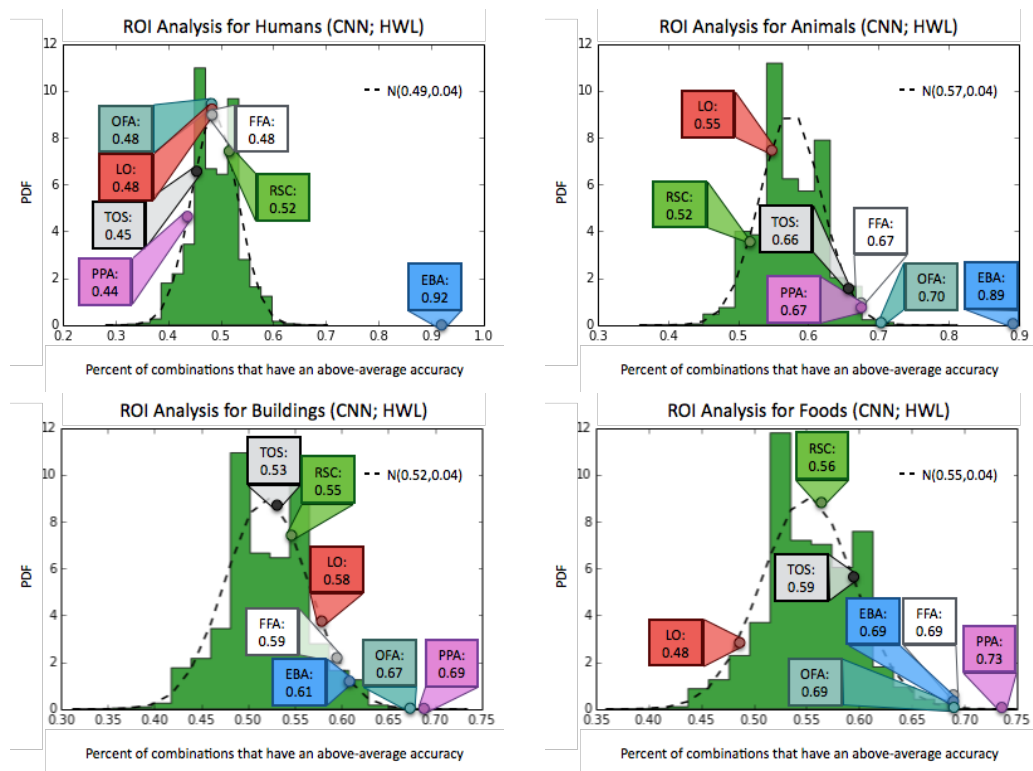


Figure 4.11: Null distributions for analysis of ROIs (CNN). See the caption for Fig. 4.10.

4.5.2 Bonferroni Confidence Intervals

Because this analysis involves comparing 127 quantities, to avoid multiple comparisons and control for the familywise error rate, I used Bonferroni correction to adjust my confidence intervals. To create m individual confidence intervals with a collective confidence interval of $1 - \alpha$, the adjusted confidence intervals can be calculated with Equation 4.4:

$$1 - \frac{\alpha}{m} \quad (4.4)$$

In my analysis, there are 127 ROI combinations, I used Equation 4.4 first with $m = 127$, $\alpha = 0.01$ and then with $m = 127$, $\alpha = 0.05$ to get the following, respective 99% and 95% Bonferroni-adjusted confidence intervals (b.c.i.s):

$$99\% \text{ b.c.i} = 1 - \frac{.01}{127} = 0.999921 \quad (4.5)$$

$$95\% \text{ b.c.i} = 1 - \frac{.05}{127} = 0.999606 \quad (4.6)$$

With these adjusted confidence intervals, I then compared outputs of the empirical CDF function $F_X(x)$ for each null distribution X that corresponded to an object category and set of image features as well as each ROI. I generated the empirical CDF functions for each null distribution using the `statsmodels.tools.tools.ECDF` function.

Let's reconsider the example of analyzing the influence of the PPA region in the classification of BUILDINGS when using CNN features. Under these experimental conditions, recall that the PPA count as a percentage is given by $x = 0.6875$. Then, output of the empirical CDF of the BUILDINGS null distribution can be calculated as $F_X(0.6875) = 0.999929$, which is greater than the adjusted 99% c.i. = 0.999921 (Eqn. 4.5).

Figures 4.12 and 4.13 show which ROIs significantly differed from the respective null distributions for each object category. This analysis confirms more rigorously the significant impacts of the EBA region in improving the classification of HUMANS and ANIMALS and of the PPA improving the classification of BUILDINGS and FOODS. Most notably, the EBA area dramatically exceeds the significance thresholds of the HUMANS and ANIMALS null distributions.

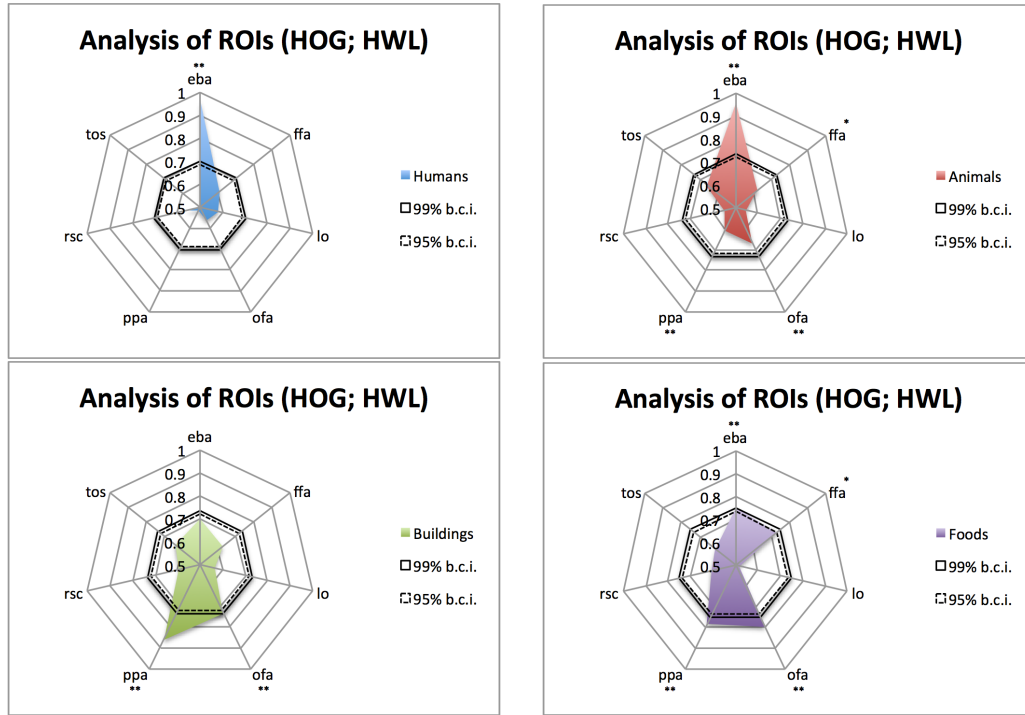


Figure 4.12: Analysis of ROIs (HOG). In each radar graph, the percent of the 64 ROI combinations containing a specific ROI (i.e. EBA) that had a mean classification accuracy greater than that over all 127 sets of experiments is plotted. These values are the same as the ones plotted in Figure 4.9 in a bar graph. The thresholds of the 99% and 95% Bonferroni-adjusted confidence intervals (b.c.i.s) are also overlayed. If the statistic for a given ROI exceeded one of the b.c.i. thresholds, appropriate significance stars were plotted by the ROI's name. See section 4.5.2 for more details about how significance testing was conducted.

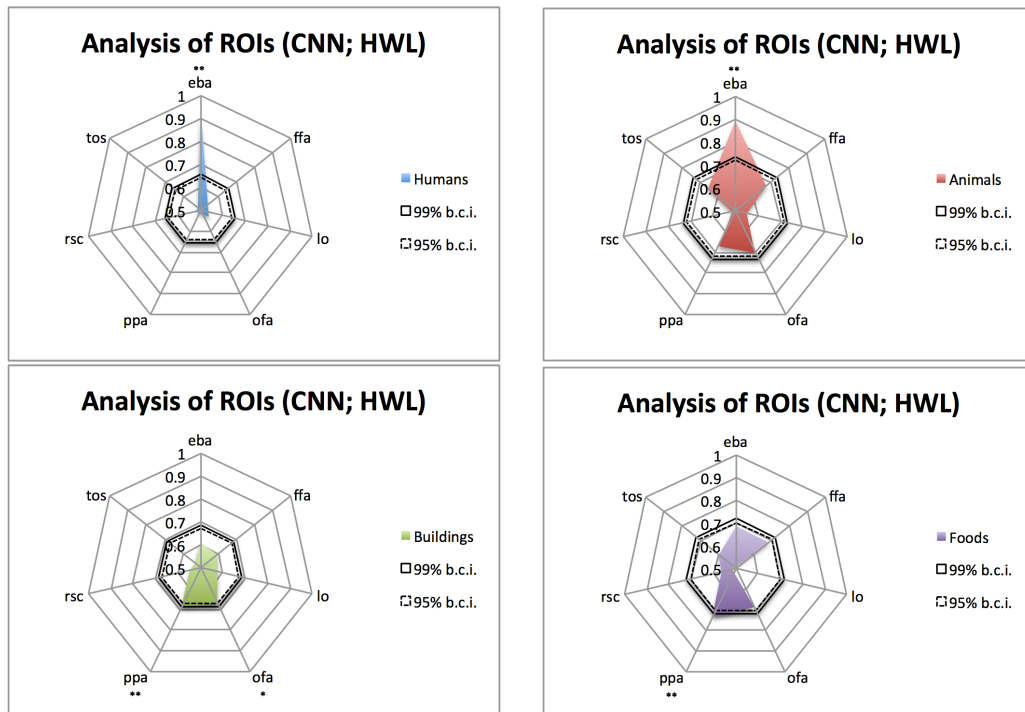


Figure 4.13: Analysis of ROIs (CNN). See the caption for Fig. 4.12.

4.6 Comparison of Different Perceptual Annotation Weighting Schemes

Throughout this thesis, I presented results in which Platt probabilities were directly used as perceptual annotations. I also ran experiments in which I first transformed Platt probabilities using Equation 4.7. The transformed annotations penalizes the misclassification of both strong positive and strong negative datum.

$$x' = |x - 0.5| \tag{4.7}$$

However, Figures 4.14 and 4.15 shows that using Platt probabilities directly as perceptual annotations is more effective than using the transformed Platt probabilities as annotations. This observation can most likely be explained by the fact that there are many more negative data points than positive ones in the dataset; thus, perceptual annotations are particularly advantageous for re-weighting strong positive data points to mitigate the effects of sparse positive data.

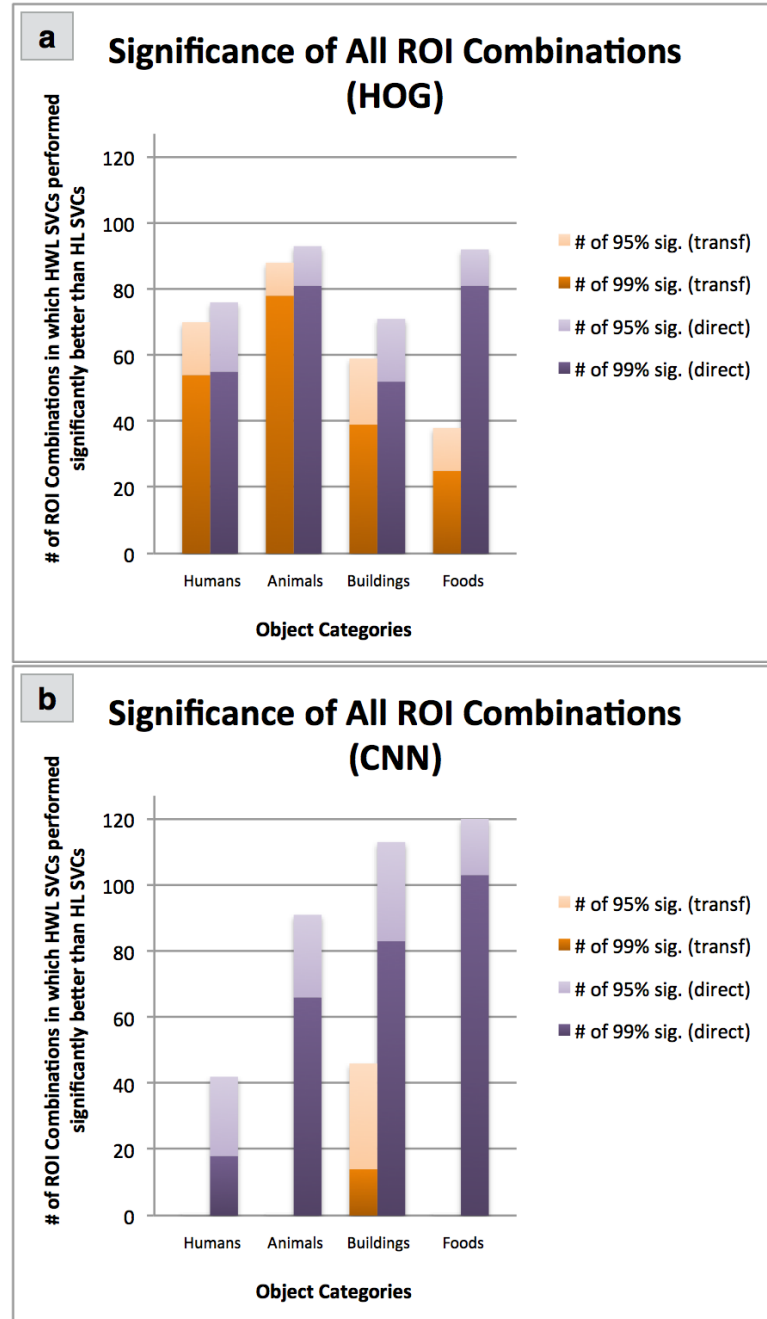


Figure 4.14: Number of significance ROI combinations using direct and transformed Platt annotations. These figures demonstrate that experiments that used Platt probabilities directly as perceptual annotations consistently had more significant improvements than those that first transformed the probabilities using Eq. 4.7.

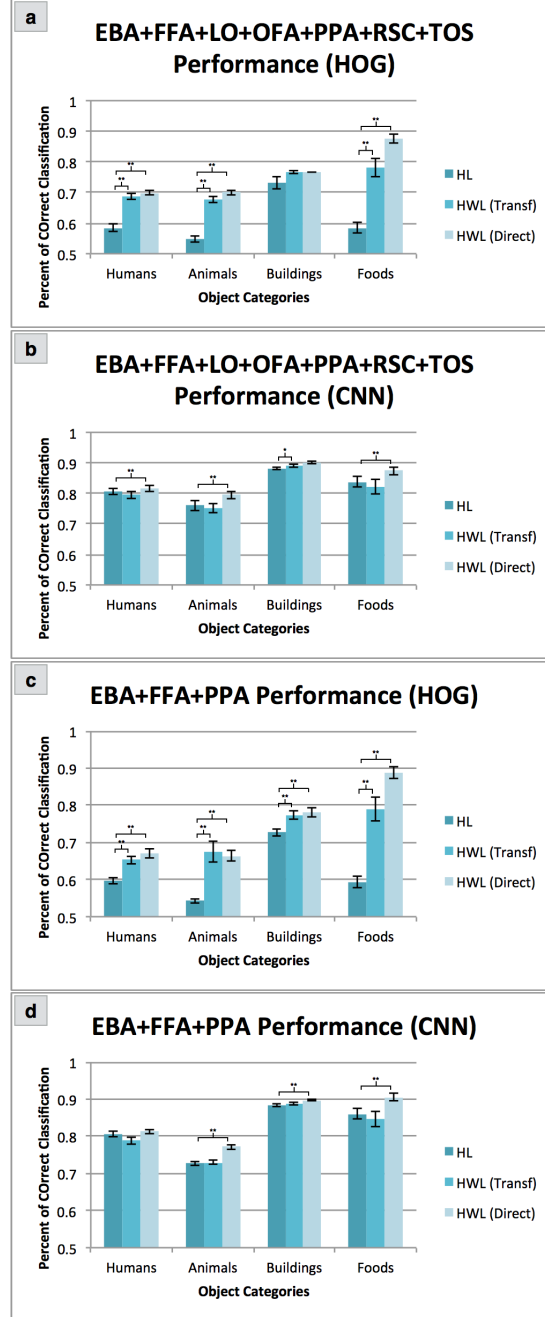


Figure 4.15: Results comparison between direct and transformed Platt annotations. Figs. 4.15b and 4.15d show clearly that experiments that used Platt probabilities directly as perceptual annotations had more significant improvements than those that first transformed the probabilities using Eq. 4.7.

Chapter 5

Discussion

In this chapter, I will discuss the advantages and limitations of using brain activity from fMRI scans as perceptual annotations to improve performance in machine vision tasks. I will also describe exciting extensions and potential future directions of the research explain in this thesis.

5.1 fMRI-based Perceptual Annotations

The results and analysis presented in Chapter 4 demonstrate how perceptual annotations based on voxel activity significantly improve the classification accuracy of SVCs in the classification of four object categories. Furthermore, section 4.5 shows how fMRI-based perceptual annotations corroborate and add to the current body of research on cortical functions.

However, there are several limitations of perceptual annotations based on brain activity. One is scalability. Like most modes of neuroimaging, fMRI data collection is difficult and time intensive. Another is the lack of detail in fMRI recordings. The strength of fMRI techniques and analyses lies in their ability to examine the interplay between different regions of the brain. Yet this technology measures blood flow and activity in large voxel units, thereby failing to capture more detail activity of individual neurons. Lastly, the current perceptual annotation technique, which uses a modified SVM with a human-weighted loss (HWL) function, requires scalar values as perceptual annotations. One significant benefit

of neuroimaging data is its richness; however, this constraint mitigates the details captured in the highly-dimensional data.

In the following sections, I suggest further work that may help overcome these issues.

5.1.1 Extension: Multi-dimensional Perceptual Annotations

One major improvement to research in perceptual annotations would be to allow for multi-dimensional annotations. This extension would enable annotations to capture more information from high-dimensional data sources like fMRI brain activity.

Another promising idea is perceptually annotated kernel matrices. Kernel functions capture the similarity between two data points and are used in SVMs to simulate the projection of data points into higher dimensions. To avoid re-computing the same function, most implementations of SVMs compute a kernel matrix that contains the output of a given kernel function applied to all pairs of training data points. Thus, supposing there are N training data points, perceptually annotated kernel matrices would allow an $O(N^2)$ -dimensional annotation, compared to the $O(N)$ 1-dimensional annotations allowed with the current algorithm. One way to generate such annotations would be to calculate a similarity matrix on the brain activity of each data point and incorporate that similarity matrix into the kernel matrix.

5.1.2 Extension: Generating Perceptual Annotations from Other Sources

Another extension of this work would be to generate perceptual annotations from other kinds of neuroimaging data, such as two-photon microscopy (Denk et al., 1990) and multi-electrode array recordings (Potter et al., 2006), which directly measures neuronal activity. The ability to directly compare the performance of perceptual annotations from different forms of neuroimaging as well as psychometric data —like the response time and accuracy of humans completing visual

experiments as done in [Scheirer et al. \(2014\)](#) —could better inform our understanding of how coarseness of perceptual annotations impacts performance on a variety of visual tasks.

Further research on perceptual annotations from the brain activity of other animal models may also provide novel insights into how visual processing varies among species. Within the neurobiology research community, substantial vision research has been conducted on animal models like monkeys and rats, which, unlike primates, have no distinct cortical regions dedicated to different kinds of visual processing. Comparing the utility of perceptual annotations between animal models with visual systems of differing complexities could further our understanding of the limitations of different species’ visual processing systems.

In addition to comparing the performance of all annotated methods using a variety of sources from which to generate perceptual annotations, another future direction involves combining the information encoded in such different modalities into a single set of annotations. One way to combine information from multiple sources lies in Representational Dissimilarity Matrices (RDMs). [Kriegeskorte et al. \(2008a\)](#) introduced RDMs as a modality-independent representation of information that could relate information encoded in brain-activity data, behavioral data, and computational models. In the space of machine learning, [Yamins et al. \(2013\)](#) presented a procedure that selects a biologically-inspired hierarchical model for object categorization that generates RDMs similar to those created by neuroimaging data from the human ventral cortical regions and the macaque IT cortex. For further research in perceptual annotations, RDMs may prove useful as a means to synthesize and represent information from multiple modalities. Such RDMs could then be used to generate 1-dimensional perceptual annotations using a method similar to the one described in section 3.2.3. Alternatively, those RDMs could be used as a multi-dimensional annotation in way comparable to the suggestion in section 5.1.1 of how similarity matrices be used as multi-dimensional annotations.

5.2 Moving towards state-of-the-art, CNN-like performance with HOG

Beyond simply improving algorithmic performance in challenging machine learning and computer vision problems, fMRI-based perceptual annotations may potentially impact research in efficient computing. In section 4.3.1, I highlight how annotations based on brain activity significantly narrowed the performance gap between HOG descriptors and CNN features in parallel object classification experiments.

Currently, CNN features are among the most state-of-the-art knowledge representation models in machine learning and have been applied to other domains quite successfully. For instance, in 2012 the clear winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was a team that used a large and deep CNN (Russakovsky et al., 2014). Two years later, nearly all teams used a CNN-based approach in ILSVRC 2014. However, large, deep CNNs require significant resources to train and use due to the computational intensity of stochastic gradient descent needed in backpropagation (see section 2.4.2 for more details about CNNs). This disadvantage of CNNs has motivated research in mimicking CNNs' performance with only a fraction of the effort: Hernández-Lobato and Adams (2015) introduces a Bayesian approach to CNNs that mitigates some of the disadvantages of stochastic optimization yet produces similar results. Similarly motivated, Hinton et al. (2014) presents techniques to compress and distill the information captured by an ensemble of CNNs into a single model.

With strong connections to robotics research and wearable technologies like Google Glass, the computer vision community has vested interests in developing computationally tractable methods that achieve state-of-the-art results. In contrast to CNNs, off-the-shelf features like HOG are highly efficient. The results presented in section 4.3.1 imply that perceptual annotations may more significantly improve basic image descriptors like HOG compared to the most cutting edge features like CNN. The empirical improvement that came from using perceptual annotations of at least halving the performance gap between HOG and CNN, as shown in Fig. 4.1, suggests the possibility that improved perceptual annotations might be able to boost the performance of off-the-shelf descriptors,

such as HOG, to be more competitive with that of CNNs.

5.2.1 Extension: CNN-based Perceptual Annotations for HOG)

To that end, in addition to improving how perceptual annotations are generated and used, incorporating CNN-based perceptual annotations in a classifier for HOG features may more directly elucidate the potential and limitations of annotated HOG descriptors. [Ba and Caruana \(2014\)](#) demonstrates how shallow CNNs can be trained to mimic deeper ones and achieve similar state-of-the-art performance. In a similar vein, perceptual annotations may be able to capture the improved feature representation of CNNs by representing qualities of the last layer of a CNN. More simply, the algorithm used to generate perceptual annotations presented in section [3.2.3](#) can be applied to CNN image features instead of fMRI brain activity data.

This additional work would still require a trained CNN in the training phase to generate annotations and train a perceptually annotated model. Nonetheless, part of the novelty of this technique lies in the fact that annotations are only needed in the training stage. Thus, annotated models can be used for deployment after training. For instance, one could train an SVC that uses CNN-based perceptual annotations to classify HOG features when computational resources are more available and then deploy the annotated classifier in wearable technologies. This extension would glean some of the benefits of rich representations in CNNs without requiring intensive computation in deployment.

5.3 Other Decision-Making Algorithms

A final area of future work includes extending the concept of perceptual annotations to other decision-making algorithms. The perceptual annotation algorithm presented in this thesis uses a modified SVM with an HWL function. However, other discriminative classifiers like logistic regression provide competitive alternatives to SVMs ([Jordan, 2002](#)). Additionally, deformable parts models ([Felzenszwalb et al., 2008](#)) and CNNs are two, state-of-the-art computer vision

algorithms for object recognition tasks. The extension of perceptual annotations to any of these methods would provide a platform for comparing how perceptual annotations performance varies between different kinds of decision-making algorithms.

Chapter 6

Conclusion

In this thesis, I presented a novel paradigm for improving decision-making methods that leverages human brain activity to force algorithms to select solutions more similar to the ones that humans would choose. Much work has been done in the way of biologically-inspired machine learning research, particularly in the area of feature representation, as evidenced by the popularity and success of convolutional neural networks. Yet, this thesis proposes an original way to use brain activity collected from fMRI scans to inform the decision-making process. Because humans continue to outperform the most state-of-the-art algorithms in challenging computer vision tasks such as object recognition, the demonstrated success of this thesis suggests that biologically-informed decision-making procedures may be critical to narrowing the gap between human and computer performance.

Earlier, I proposed a novel algorithm for generating salient, scalar-value perceptual annotations from high-dimensional brain activity that capture the “learnability” of an image for an object classification task. I then outlined how to use a modified SVM classifier to more aggressively penalize the misclassification of training data based on their annotations with a “human-weighted” loss function. This procedure weighted the correct classification of images that annotations suggest were easily “learnable” over the classification of images that the human brain struggled with.

Then, I presented empirical results demonstrating the significant, comparative improvements of fMRI-based perceptual annotations on object classification

tasks. I also showed how such annotations empirically halved the difference in classification performance between HOG and CNN image features. This impressive result suggests that perceptual annotations may be able to boost the performance of efficient, off-the-shelf feature representation algorithms like HOG to be competitive with that of the state-of-the-art, yet computationally intensive CNNs. Through further analysis, I demonstrated how certain cortical regions contributed to specific, large gains in classification performance for different object categories, such as how the inclusion of voxels from EBA significantly improved HUMANS classification. These results affirmed the current body of neuroimaging literature on cortical functions.

Finally, I outlined various extensions to this work that could continue to push boundaries in the areas of perceptual annotation, efficient computing, and decision-making. Such research could pave the way to artificial intelligence that is truly competitive with our own.

Appendix A

To deal with the scarcity of data, I focused on classifying images as including objects from the following five categories: humans, animals, buildings, foods, and vehicles. Each pixel of each of the 1386 images from 2013 Stansbury et. al was manually labelled with one of 1387 object labels. I mapped 271 of those 1387 original object labels to the five object categories I was interested in. Below are those mappings.

To deal with the scarcity of data, I focused on classifying images as including objects from the following five categories: humans, animals, buildings, foods, and vehicles. Each pixel of each of the 1386 images from [Stansbury et al. \(2013\)](#) was manually labelled with one of 1387 object labels. I mapped 271 of those 1387 original object labels to the five object categories I was interested in. Below are those mappings.

I mapped the following 16 object labels from [Stansbury et al. \(2013\)](#) to the object category HUMANS: human, person, man, woman, humans, boy, girl, crowd, people, referee, athlete, audience, men, women, girls, boys.

I mapped the following 165 object labels from [Stansbury et al. \(2013\)](#) to the object category ANIMALS: abalone, alligator, amphibian, animal, animals, antelope, ants, ape, bear, bears, bees, beetle, bird, birds, butterflies, butterfly, buzzard, calf, camel, camels, cat, cattle, cheetah, chicken, chickens, chimpanzee, cobra, cobras, cougar, cow, cow_legs, cows, coyote, crab, deer, dog, dogs, dolphin, dolphins, donkey, duck, ducks, eagle, eel, elelphant, elephant, elephant_seal, elephants, exotic_fish, fish, fish_head, fish_tail, flatfish, fox, frog,

gazelles, gecko, giraffe, goat, goats, groundhog, hawk, hippopotamus, hiyena, horse, horse_ass, horses, hounds, iguana, insect, kangaroo, killer_whale, koala, ladybug, land_mammal, lemur, lion, lioness, lions, lizard, lizard_face, lizards, llama, llamas, lobster, lobsters, mammal, mammals, monkey, moose, moray_eel, moth, moths, mountain_goat, mouse, mutt, newt, octopus, ostrich, ostriches, otter, ox, oxen, panda, parrot, penguin, penguins, pheasant, pig, pigeon, pigeons, pony, porcupine, prairie_dog, prairie_dogs, praying_mantis, primate, raccoon, racoon, rattlesnake, reptile, rhinoceros, rodent, rodents, salamander, sea_lion, sea_lions, sea_turtle, seagull, seal, seal_ass, seal_pup, seals, shark, sharks, sheep, skunk, snake, snakes, spider, spiders, squirrel, stag, stingray, stork, storks, submersed_animal, swordfish, tiger, tigers, toad, tortoise, tortoises, tuna, turkey, turtle, turtle_shell, walrus, water_buffalo, whale, wild_boar, yak, yaks, zebra, zebras.

I mapped the following 4 object labels from [Stansbury et al. \(2013\)](#) to the object category BUILDINGS: building, buidling, buildings, building.

I mapped the following 58 object labels from [Stansbury et al. \(2013\)](#) to the object category FOODS: apples, baby_food, banannas, berries, blackberries, cabbage, cantelope, cheese, cherries, corn, croutons, french_fries, fries, fruit, garlic, garlic_cloves, ginger, gourd, grains, grapes, green_pepper, hamburger, kiwi, lemon, lemons, lettuce, meat, melon, nuts, okra, oranges, pastry, payapas, peas, pepper, peppers, pie, pistachios, pizza, potatoes, pumpkins, purple_potatoes, red_pepper, red_potatoes, rice, root, salad, saurkraut, sausage, squash, strawberries, toast, tomatoes, vegetable, vegetables, waffle, watermelon, watermelons.

I mapped the following 28 object labels from [Stansbury et al. \(2013\)](#) to the object category VEHICLES: airplane, boat, boats, bulldozer, bus, car, cars, cruiser, fire_engine, fishing_boat, four_wheeler, minibus, ship, snowmobile, speedboat, steamboat, tank, taxi, tow_truck, tractor, train, trains, truck, van, vehicle, vehicles, wagon, yacht.

Appendix B

Below are the 127 combinations of the 7 higher-level visual cortical regions that I used in my experiments as well as the number of voxels that are in each combination:

EBA (322 voxels), FFA (179 voxels), LO (160 voxels), OFA (124 voxels), PPA (264 voxels), RSC (205 voxels), TOS (173 voxels), EBA+FFA (501 voxels), EBA+LO (482 voxels), EBA+OFA (446 voxels), EBA+PPA (586 voxels), EBA+RSC (527 voxels), EBA+TOS (495 voxels), FFA+LO (339 voxels), FFA+OFA (303 voxels), FFA+PPA (443 voxels), FFA+RSC (384 voxels), FFA+TOS (352 voxels), LO+OFA (284 voxels), LO+PPA (424 voxels), LO+RSC (365 voxels), LO+TOS (333 voxels), OFA+PPA (388 voxels), OFA+RSC (329 voxels), OFA+TOS (297 voxels), PPA+RSC (469 voxels), PPA+TOS (437 voxels), RSC+TOS (378 voxels), EBA+FFA+LO (661 voxels), EBA+FFA+OFA (625 voxels), EBA+FFA+PPA (765 voxels), EBA+FFA+RSC (706 voxels), EBA+FFA+TOS (674 voxels), EBA+LO+OFA (606 voxels), EBA+LO+PPA (746 voxels), EBA+LO+RSC (687 voxels), EBA+LO+TOS (655 voxels), EBA+OFA+PPA (710 voxels), EBA+OFA+RSC (651 voxels), EBA+OFA+TOS (619 voxels), EBA+PPA+RSC (791 voxels), EBA+PPA+TOS (759 voxels), EBA+RSC+TOS (700 voxels), FFA+RSC+TOS (557 voxels), FFA+PPA+TOS (616 voxels), FFA+PPA+RSC (648 voxels), FFA+OFA+TOS (476 voxels), FFA+OFA+RSC (508 voxels), FFA+OFA+PPA (567 voxels), FFA+LO+TOS (512 voxels), FFA+LO+RSC (544 voxels), FFA+LO+PPA (603 voxels), FFA+LO+OFA (463 voxels), LO+RSC+TOS (538 voxels), LO+PPA+TOS (597 voxels), LO+PPA+RSC (629 voxels), LO+OFA+TOS (457 voxels), LO+OFA+RSC (489 voxels),

LO+OFA+PPA (548 voxels), OFA+PPA+RSC (593 voxels), OFA+PPA+TOS (561 voxels), OFA+RSC+TOS (502 voxels), PPA+RSC+TOS (642 voxels), OFA+PPA+RSC+TOS (766 voxels), LO+PPA+RSC+TOS (802 voxels), LO+OFA+RSC+TOS (662 voxels), LO+OFA+PPA+TOS (721 voxels), LO+OFA+PPA+RSC (753 voxels), FFA+PPA+RSC+TOS (821 voxels), FFA+OFA+RSC+TOS (681 voxels), FFA+OFA+PPA+TOS (740 voxels), FFA+OFA+PPA+RSC (772 voxels), FFA+LO+RSC+TOS (717 voxels), FFA+LO+PPA+TOS (776 voxels), FFA+LO+PPA+RSC (808 voxels), FFA+LO+OFA+TOS (636 voxels), FFA+LO+OFA+RSC (668 voxels), FFA+LO+OFA+PPA (727 voxels), EBA+PPA+RSC+TOS (964 voxels), EBA+OFA+RSC+TOS (824 voxels), EBA+OFA+PPA+TOS (883 voxels), EBA+OFA+PPA+RSC (915 voxels), EBA+LO+RSC+TOS (860 voxels), EBA+LO+PPA+TOS (919 voxels), EBA+LO+PPA+RSC (951 voxels), EBA+LO+OFA+TOS (779 voxels), EBA+LO+OFA+RSC (811 voxels), EBA+LO+OFA+PPA (870 voxels), EBA+FFA+RSC+TOS (879 voxels), EBA+FFA+PPA+TOS (938 voxels), EBA+FFA+PPA+RSC (970 voxels), EBA+FFA+OFA+TOS (798 voxels), EBA+FFA+OFA+RSC (830 voxels), EBA+FFA+OFA+PPA (889 voxels), EBA+FFA+LO+TOS (834 voxels), EBA+FFA+LO+RSC (866 voxels), EBA+FFA+LO+PPA (925 voxels), EBA+FFA+LO+OFA (785 voxels), LO+OFA+PPA+RSC+TOS (926 voxels), FFA+OFA+PPA+RSC+TOS (945 voxels), FFA+LO+PPA+RSC+TOS (981 voxels), FFA+LO+OFA+RSC+TOS (841 voxels), FFA+LO+OFA+PPA+TOS (900 voxels), FFA+LO+OFA+PPA+RSC (932 voxels), EBA+OFA+PPA+RSC+TOS (1088 voxels), EBA+LO+PPA+RSC+TOS (1124 voxels), EBA+LO+OFA+RSC+TOS (984 voxels), EBA+LO+OFA+PPA+TOS (1043 voxels), EBA+LO+OFA+PPA+RSC (1075 voxels), EBA+FFA+PPA+RSC+TOS (1143 voxels), EBA+FFA+OFA+RSC+TOS (1003 voxels), EBA+FFA+OFA+PPA+TOS (1062 voxels), EBA+FFA+OFA+PPA+RSC (1094 voxels), EBA+FFA+LO+RSC+TOS (1039 voxels), EBA+FFA+LO+PPA+TOS (1098 voxels), EBA+FFA+LO+PPA+RSC (1130 voxels), EBA+FFA+LO+OFA+TOS (958 voxels), EBA+FFA+LO+OFA+RSC (990 voxels), EBA+FFA+LO+OFA+PPA (1049 voxels),

EBA+FFA+LO+OFA+PPA+RSC (1254 voxels),
EBA+FFA+LO+OFA+PPA+TOS (1222 voxels),
EBA+FFA+LO+OFA+RSC+TOS (1163 voxels),
EBA+FFA+LO+PPA+RSC+TOS (1303 voxels),
EBA+FFA+OFA+PPA+RSC+TOS (1267 voxels),
EBA+LO+OFA+PPA+RSC+TOS (1248 voxels),
FFA+LO+OFA+PPA+RSC+TOS (1105 voxels),
EBA+FFA+LO+OFA+PPA+RSC+TOS (1427 voxels)

References

- Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014. [64](#)
- James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of The 30th International Conference on Machine Learning*, pages 115–123, 2013. [5](#)
- Christopher M. Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006. [17](#), [21](#), [24](#), [27](#)
- Anna Bosch, Andrew Zisserman, and Xavier Muoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. [6](#)
- Steve Branson, Pietro Perona, and Serge Belongie. Strong supervision from weak annotation: Interactive training of deformable part models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1832–1839. IEEE, 2011. [6](#)
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. [6](#)
- Richard B. Buxton. *Introduction to functional magnetic resonance imaging: principles and techniques*. Cambridge university press, 2009. [9](#)
- Charles F. Cadieu, Ha Hong, Daniel L.K. Yamins, Nicolas Pinto, Diego Ardila, Ethan A. Solomon, Najib J. Majaj, and James J. DiCarlo. Deep neural net-

REFERENCES

- works rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10(12):e1003963, 2014. 1, 5
- Murray Campbell, A. Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1):57–83, 2002. 1
- Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3, 2010. 2
- Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 33
- Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Trading convexity for scalability. In *Proceedings of the 23rd international conference on Machine learning*, pages 201–208. ACM, 2006. 16
- David Cox and Nicolas Pinto. Beyond simple features: A large-scale feature search approach to unconstrained face recognition. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 8–15. IEEE, 2011. 1, 5
- David D. Cox and Robert L. Savoy. Functional magnetic resonance imaging (fmri) brain reading: detecting and classifying distributed patterns of fmri activity in human visual cortex. *Neuroimage*, 19(2):261–270, 2003. 7, 40
- Tolga Çukur, Alexander G. Huth, Shinji Nishimoto, and Jack L. Gallant. Functional subdomains within human ffa. *The Journal of Neuroscience*, 33(42):16748–16766, 2013. 34
- N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, 2005. 5, 22
- DeepLearning. Convolutional neural networks (lenet). <http://deeplearning.net/tutorial/lenet.html>, 2015. Accessed online on March 30, 2015. 24

REFERENCES

- Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. [6](#)
- Winfried Denk, James H. Strickler, and Watt W. Webb. Two-photon laser scanning fluorescence microscopy. *Science*, 248(4951):73–76, 1990. [61](#)
- Hannah Devlin. What is functional magnetic resonance imaging (fmri). *Psych Central*, 2008. [9](#), [10](#)
- James J. DiCarlo and David D. Cox. Untangling invariant object recognition. *Trends in cognitive sciences*, 11(8):333–341, 2007. [16](#)
- Paul E. Downing, Yuhong Jiang, Miles Shuman, and Nancy Kanwisher. A cortical area selective for visual processing of the human body. *Science*, 293(5539):2470–2473, 2001. [11](#), [44](#)
- Chris Eliasmith. *How to build a brain: A neural architecture for biological cognition*. Oxford University Press, 2013. [5](#)
- Chris Eliasmith, Terrence C. Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *science*, 338(6111):1202–1205, 2012. [5](#)
- Russell Epstein and Nancy Kanwisher. A cortical representation of the local visual environment. *Nature*, 392(6676):598–601, 1998. [13](#), [44](#), [46](#)
- Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005. [5](#)
- Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. [64](#)

REFERENCES

- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010. 1
- Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995. 6
- Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom M. Mitchell. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *EMNLP*, pages 833–838, 2013. 6
- Isabel Gauthier, Pawel Skudlarski, John C Gore, and Adam W. Anderson. Expertise for cars and birds recruits brain areas involved in face recognition. *Nature neuroscience*, 3(2):191–197, 2000a. 11
- Isabel Gauthier, Michael J. Tarr, Jill Moylan, Pawel Skudlarski, John C. Gore, and Adam W. Anderson. The fusiform face area is part of a network that processes faces at the individual level. *Journal of cognitive neuroscience*, 12(3):495–504, 2000b. 13, 34, 44, 46
- Zoya Gavrilov. Svm tutorial. <http://web.mit.edu/zoya/www/SVM.pdf>, 2015. Accessed online on March 29, 2015. 17
- Kalanit Grill-Spector, Zoe Kourtzi, and Nancy Kanwisher. The lateral occipital complex and its role in object recognition. *Vision research*, 41(10):1409–1422, 2001. 13
- Uri Hasson, Michal Harel, Ifat Levy, and Rafael Malach. Large-scale mirror-symmetry organization of human occipito-temporal object areas. *Neuron*, 37(6):1027–1041, 2003. 13
- José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. *arXiv preprint arXiv:1502.05336*, 2015. 63

REFERENCES

- Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. [26](#), [36](#)
- Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*, 2014. [63](#)
- D.H. Hubel and T.N. Wiesel. Shape and arrangement of columns in cat’s striate cortex. *The Journal of physiology*, 165(3):559–568, 1963. [4](#)
- Scott A. Huettel, Allen W. Song, and Gregory McCarthy. *Functional magnetic resonance imaging*, volume 1. Sinauer Associates Sunderland, 2004. [9](#)
- Alexander G. Huth, Shinji Nishimoto, An T. Vu, and Jack L. Gallant. A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron*, 76(6):1210–1224, 2012. [7](#)
- Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. [24](#), [31](#)
- Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841, 2002. [64](#)
- Nancy Kanwisher, Josh McDermott, and Marvin M. Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *The Journal of Neuroscience*, 17(11):4302–4311, 1997. [11](#), [34](#), [44](#), [46](#)
- Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352–355, 2008. [7](#), [29](#)
- Nikolaus Kriegeskorte, Elia Formisano, Bettina Sorger, and Rainer Goebel. Individual faces elicit distinct response patterns in human anterior temporal cortex. *Proceedings of the National Academy of Sciences*, 104(51):20600–20605, 2007. [40](#)

REFERENCES

- Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis—connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2, 2008a. [7](#), [62](#)
- Nikolaus Kriegeskorte, Marieke Mur, Douglas A. Ruff, Roozbeh Kiani, Jerzy Bodurka, Hossein Esteky, Keiji Tanaka, and Peter A. Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141, 2008b. [8](#)
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [31](#), [32](#)
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. [4](#)
- Li-Jia Li and Li Fei-Fei. What, where and who? classifying events by scene and object recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. [6](#)
- Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P. Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in neural information processing systems*, pages 1378–1386, 2010. [5](#)
- Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on plat’s probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007. [21](#)
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014*, pages 740–755. Springer, 2014. [3](#)

REFERENCES

- David G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. [5](#)
- Eleanor Maguire. The retrosplenial contribution to human navigation: a review of lesion and neuroimaging findings. *Scandinavian journal of psychology*, 42(3):225–238, 2001. [13](#)
- Ronald P.S. Mahler. *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007. [44](#)
- Rafael Malach, J.B. Reppas, R.R. Benson, K.K. Kwong, H. Jiang, W.A. Kennedy, P.J. Ledden, T.J. Brady, B.R. Rosen, and R.B. Tootell. Object-related activity revealed by functional magnetic resonance imaging in human occipital cortex. *Proceedings of the National Academy of Sciences*, 92(18):8135–8139, 1995. [13](#)
- Jeff Miller. Divided attention: Evidence for coactivation with redundant signals. *Cognitive psychology*, 14(2):247–279, 1982. [6](#)
- Tom M. Mitchell, Svetlana V. Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L. Malave, Robert A. Mason, and Marcel Adam Just. Predicting human brain activity associated with the meanings of nouns. *science*, 320(5880):1191–1195, 2008. [7](#)
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. [5](#)
- Marieke Mur, Peter A. Bandettini, and Nikolaus Kriegeskorte. Revealing representational content with pattern-information fmri introductory guide. *Social cognitive and affective neuroscience*, page nsn044, 2009. [7](#)
- K. Nakamura, R. Kawashima, N. Sato, A. Nakamura, M. Sugiura, T. Kato, K. Hatano, K. Ito, H. Fukuda, T. Schormann, et al. Functional delineation of the human occipito-temporal areas related to face and scene processing a pet study. *Brain*, 123(9):1903–1912, 2000. [13](#)

REFERENCES

- Thomas Naselaris, Ryan J. Prenger, Kendrick N. Kay, Michael Oliver, and Jack L. Gallant. Bayesian reconstruction of natural images from human brain activity. *Neuron*, 63(6):902–915, 2009. 7
- Shinji Nishimoto, An T. Vu, Thomas Naselaris, Yuval Benjamini, Bin Yu, and Jack L. Gallant. Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, 21(19):1641–1646, 2011. 7
- Jin Park. Unconsciousness: How fmri allows the brain to examine consciousness, unconsciousness, and free will. <http://jinpark.egloos.com/v/821655>, 2011. Accessed March 29, 2015. 11
- Avi Pfeffer, David Parkes, and Ryan P. Adams. Max-margin classification. Harvard Computer Science 181 Course Notes, 2014. <http://isites.harvard.edu/fs/docs/icb.topic1362126.files/maxmargin.pdf>. 17, 18
- Nicolas Pinto, David Doukhan, James J. DiCarlo, and David D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS computational biology*, 5(11):e1000579, 2009. 1, 5
- John Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, and B. Schölkopf, editors, *Advances in Large Margin Classifiers*. MIT Press, 2000. 21, 33
- Russell A. Poldrack. Region of interest analysis for fmri. *Social cognitive and affective neuroscience*, 2(1):67–70, 2007. 11
- S.M. Potter, D.A. Wagenaar, T.B. DeMarse, M. Taketani, and M. Baudry. Advances in network electrophysiology using multi-electrode arrays, 2006. 61
- J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986. 5
- Leila Reddy, Naotsugu Tsuchiya, and Thomas Serre. Reading the mind’s eye: decoding category information during mental imagery. *Neuroimage*, 50(2):818–825, 2010. 7

REFERENCES

- Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999. [4](#)
- Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 1996. [24](#)
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet Large Scale Visual Recognition Challenge, 2014. [1](#), [3](#), [43](#), [63](#)
- Walter Scheirer, Samuel E. Anthony, Ken Nakayama, and David D. Cox. Perceptual annotation: Measuring human vision to improve computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2014. [2](#), [3](#), [13](#), [14](#), [15](#), [17](#), [33](#), [62](#)
- Thomas Serre, Lior Wolf, and Tomaso Poggio. Object recognition with features inspired by visual cortex. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 994–1000. IEEE, 2005. [5](#)
- Thomas Serre, Gabriel Kreiman, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, and Tomaso Poggio. A quantitative theory of immediate visual recognition. *Progress in brain research*, 165:33–56, 2007a. [4](#)
- Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104(15):6424–6429, 2007b. [4](#)
- Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):411–426, 2007c. [4](#)
- Svetlana V. Shinkareva, Robert A. Mason, Vicente L. Malave, Wei Wang, Tom M. Mitchell, and Marcel Adam Just. Using fmri brain activation to identify cog-

REFERENCES

- nitive states associated with perception of tools and dwellings. *PLoS One*, 3(1):e1394, 2008. 7
- Alex J. Smola and Bernhard Schölkopf. *Learning with kernels*. Citeseer, 1998. 14
- Mona Spiridon, Bruce Fischl, and Nancy Kanwisher. Location and spatial profile of category-specific regions in human extrastriate cortex. *Human brain mapping*, 27(1):77–89, 2006. 28, 40
- Dustin E. Stansbury, Thomas Naselaris, and Jack L. Gallant. Natural scene statistics account for the representation of scene categories in human visual cortex. *Neuron*, 79(5):1025–1034, 2013. 7, 12, 28, 29, 30, 68, 69
- Zak Stone, Todd Zickler, and Trevor Darrell. Autotagging facebook: Social network context improves photo annotation. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW’08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008. 3, 6
- Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033*, 2011. 6
- Antonio Torralba, Aude Oliva, Monica S. Castelhana, and John M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006. 6
- UFLDL. Convolutional neural network. <http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>, 2015. Accessed online on March 30, 2015. 24, 27
- Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 2
- A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008. 23, 31
- Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Computer Vision*

REFERENCES

- and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2798–2805. IEEE, 2014. [5](#)
- Stephanie Watson. How fmri works. *How Stuff Works*, 2011. [9](#), [10](#)
- Peter Welinder, Steve Branson, Pietro Perona, and Serge J. Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010. [6](#)
- Daniel L. Yamins, Ha Hong, Charles F. Cadieu, and James J. DiCarlo. Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. In *Advances in Neural Information Processing Systems*, pages 3093–3101, 2013. [5](#), [62](#)
- Daniel L.K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014. [5](#)
- Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014. [25](#)