Draw, Don't Look: Motor Representations for Few-shot Object Recognition

A project report submitted in partial fulfilment of the requirements

for the degree of Master of Science (by Research)

by

Aastha Sharma

DEPARTMENT OF COGNITIVE SCIENCE

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

August 2021

Certificate

It is certified that the work contained in this project report entitled '*Draw, Don't Look*: Motor Representations for Few-shot Object Recognition' by Aastha Sharma has been carried out under my supervision and that it has not been submitted elsewhere for a degree.

Dr. Nisheeth Srivastava

Assistant Professor

Department of Cognitive Science

Indian Institute of Technology Kanpur

August 2021

Declaration

This is to certify that the project report entitled **'*Draw, Don't Look*: Motor Representations for Few-shot Object Recognition'** has been authored by me. It presents research conducted by me under the supervision of Dr. Nisheeth Srivastava. To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted elsewhere, in part or in full, for a degree. Further, due credit has been attributed to the relevant state-of-the-art and collaborations (if any) with appropriate citations and acknowledgements, in line with established norms and practices.

Aastha Sharma

Master of Science (by Research)

Department of Cognitive Science

Indian Institute of Technology Kanpur

Abstract

Name of the student: **Aastha Sharma** Roll No: **18128401**

Degree for which submitted: **MS(by Research)** Department: **Cognitive Science**

Project title: ***Draw, Don't Look*: Motor Representations for Few-shot Object Recognition**

Project supervisor: **Dr. Nisheeth Srivastava**

Month and year of project submission: **August 2021**

Machines do not look at the world the way humans do - humans generalize rapidly and accurately, while machines need extensive training. Machines recognize zebra crossings as actual zebras and school buses as giraffes, and humans do not. We propose that artificial agents can be brought closer to humans by improved representation learning. Specifically, we hypothesize that learning motor representations instead of purely visual ones can improve visual similarity judgments. We investigate this by embedding line drawings of objects in a motor space and computing similarities. We use a Program Learning paradigm to infer motor programs and define and test novel metrics of motor program similarity. Our formulation of motor programs and motor similarity fails to produce human-like category discrimination, and we conclude by presenting potential rectifications for the same.

Acknowledgements

This project is, by no means, the work of an individual. I consider myself fortunate to have had the support of many people in this process. Starting, of course, with Nisheeth Srivastava, my supervisor, who taught me to be critical and think about things deeply. I am grateful for his support and mentorship and have come to the realization that, despite our philosophical differences, he has influenced the way I think about life.

My fellow Y18 members – Aditi, Akshay, Anjoom, Harish, Komal, Neelabja, Oviya, Rithwik, and Smith – who made the last three years the most enjoyable three years of my life. It is rare to find a friend as insane as one's own self, but to find nine of them? That's a lottery. Perhaps one day we will all agree on UNO rules and create '*Community: CGS Version*'.

Arjun Mitra and Pratham Shukla let me go on and on about the insignificant problems of my life and somehow continue to be my friends.

Shrutika Jha, Rithwik Cherian, Aastik Ahuja, and Shruti Lahiry – who have been selfless in their love for as long as I have known them.

The people of IIT-Kanpur – its workers, students, professors - who made it my home for three years, in every way possible.

Finally, and most importantly, my parents – who have tolerated me for over two decades. I am because you are.

को अद्धा वेद क इह प्र वोचत्कुत आजाता कुत इयं विसृष्टि:।
अर्वाग्देवा अस्य विसर्जनेनाथा को वेद यत आबभूव।।

Who really knows? Who will here proclaim it?
Whence was it produced? Whence is this creation?
The gods themselves are later than creation,
so who knows truly whence it has arisen?

Nasadiya Sukta, Rigveda 10:129

Contents

ListofFigures

Chapter 1

Introduction

1.1 Introduction

Consider the following image of a tapir.

FIGURE 1.1: The tapir is a large, pig-like, herbivorous mammal

Now look at the following 6 images and label them as tapir or non-tapir.

FIGURE 1.2: Label these images as tapir or non-tapir

Chances are that you were able to label the images accurately (only the first 3 are tapirs) and effortlessly. Assuming that you hadn't seen a tapir before, you just learnt to recognize one through a single image. Not only that, you generalized what you learnt across size,

viewpoint and colour. Learning and generalizations of this kind are considered hallmarks of intelligent behaviour. While it comes naturally to humans, a neural network trained on the same image fails at the classification task. What is it that we learn and machines don't? And can they be taught to learn like us?

In figure 1.1, you saw a number of features - color, snout, four legs, shape and size of the body, etc. The anteater, pig and aardvark from 1.2 shared some of these features, but didn't quite look like a tapir. At the same time, the first three photos lacked some features but you recognized them (presumably) correctly. You learnt what it means for something to be a tapir; its *tapir-ness*, so to say. When we see an object, we represent it in a format that allows mental manipulation. In other words, we create a *representation*. Among other things, we use the representation to deduce the object-*ness* of the object, or what it means for something to be that object. This deduction leads to the formation of a *concept* of the object. Although cognitive scientists are divided vis-a-vis the exact nature of representations and concepts, most of them agree that they exist in the mind and aid recognition and categorization.

The neural network that sees figure 1.1 also creates a representation in the form of an activation pattern and weight parameters. However, it faces a data-versus-generalization tradeoff. With a single data point, its representation is too specific. That is, an object must possess all the features of the exemplar to classify as a tapir. In order to make the representation generalized, the neural network needs more data points. This tradeoff causes learning in machines to be data-intensive and non-general.

1.2 Objective

The key objective of this work is to enable few-shot, transformation-invariant learning of visual concepts in machines. Not only should they learn from a small amount of data, they should also recognize objects as they undergo transformations (such as changes in object position, distance, pose).

Instead of building on standard ML approaches, we take cues from human concept-learning and object recognition, and attempt to build an intuitive and cognitively informed model.

1.3 Proposed Approach

Traditional paradigms view object recognition as a feature-matching exercise. Objects are represented as vectors in feature spaces, such that clusters of vectors constitute individual

categories. However, accounting for all possible features increases the dimensionality of vectors and consequently makes generalization impossible. Therefore, machines need to learn which features define an object or a category. Unlike humans, they iterate through millions of images to do this.

We propose comparing how objects are drawn instead of representing them only as collections of visual features. We believe that drawing patterns (called motor programs henceforth) of objects can be more discriminative than visual features alone. We use a Bayesian Learning approach to create motor representations. The Bayesian approach is also useful in reducing the data requirements and enabling few-shot learning.

1.4 Organization

We begin with building an intuitive understanding of representations and reviewing philosophical standpoints in Chapter 2. In Chapter 3, we set the stage for shape skeletons and present a case for their usefulness in representing object structure. Then we describe the Bayesian Program Learning framework, which was used for motor program inference, in 4. Chapter 5 is dedicated to our own experiments and analyses, followed by a discussion about the implications of this work and future possibilities in Chapter 6.

Chapter 2

Representations

A discussion about recognition requires being prefaced with a discussion about representations. When we say we recognize an object X, it implies that X looks like something we've seen before. That is, we compare what we see with something stored in our memory and conclude that it is a match. This matching process warrants representing both the object and our memory of it in a standard format.
Many philosophers have used computationally-inspired terms - 'symbol strings', 'information-bearing states' and 'machine tables'[47] - to describe representations. Suppes [50] gave a more concise definition - "Representation of something is an image, model, or reproduction of that thing." We can obtain a more functional characterization by asking what representations should be.

1. They should correspond to something in the world: Because they support recognition and categorization, representations should be proxies for objects *out there*.

2. They should be general, but also specific: Transformations (change in viewpoint, rotation, etc.) can change the way things appear to people. Representations should be generalized enough to account for such variations. Simultaneously, they should be discriminative (i.e., one representation should not stand for more than one object or class of objects).

3. They should be easy to learn: Humans learn object identities and novel categories rapidly, using very few examples. Representational structures that support this learning should not be complex.

4. They should be manipulatable: Cognitive processes involve manipulating information. Assuming that representations are the information-bearers, they should be

capable of being manipulated. In the specific case of recognition, it should be possible to compute similarity between two given representations.

2.1 What comprises representations?

What comprises representations? – the question has kept philosophers of mind occupied for a long time. The candidates are diverse - images, data structures, activation levels in processors. On the other end of the spectrum, proponents of the dynamical theory of mind reject the very notion of representations and propose a non-computational, dynamical systems view [24].
We divide the prospective constituents into four categories:

1. Imagery
2. Symbols
3. Feature spaces
4. Structural descriptions

2.1.1 Imagery

Thinking about things in the world usually involves picturing them. Therefore, it is intuitive to consider that representations must be images. This proposition has its roots in Aristotelian thought. 'To represent something is to be it' - this was the central tenet of what Cummins [12] calls the 'Mind-Stuff-Informed' class of theories. In these theories, representations of the world are mental models that capture all physical aspects of the real world. Such a line of thought is sometimes also called *reconstructionism* - representing the information we receive by reconstructing it as it is. Some authors [17] have claimed that Marr's early work also falls in the reconstructionist domain. Much of vision research in the 1980s was aimed at finding the best way to reconstruct the world in 3D [2][8].

FIGURE 2.1: Aristotle representing a cat with a ball by reconstructing it as it is. Source: Cummins (1991)[12]

FIGURE 2.2: Berkeley also represents a cat with a ball by reconstruction. Source: Cummins (1991)[12]

In theory, a replica of an object is the best possible candidate for visual tasks because practically no information is wasted or lost. But in practice, reconstructionism leads to a multitude of problems [18] [45].

1. Grounding

   The reconstructionist hypothesis does not explain how mental replicas can be grounded in meaning. In Figures 2.1 and 2.2, Aristotle and Berkeley represent a cat with a ball by creating mental reconstructions. But it is unclear how those reconstructions will be compared with other such reconstructions, or how Aristotle and Berkeley would derive meaning from them.

2. A representation for everything

   Generalization across transformations is difficult in image representations. If the mind had a 2D image of an object, a change in viewpoint or rotation in depth would render the mental image useless. Consequently, a person would need to create a new mental picture for every transformation of the same object. A 3D model, on the other hand, would be hard to learn due to the number of exemplars needed for model creation.

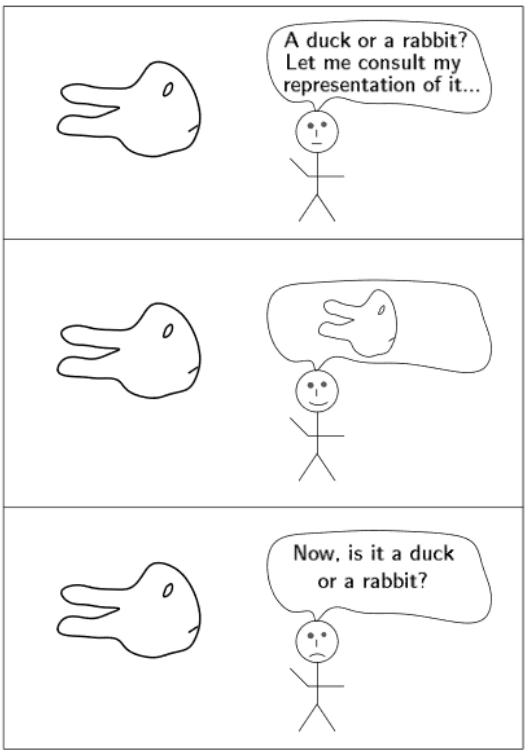


FIGURE 2.3: Jastrow's duck/rabbit drawing, used by Wittgenstein in *Philosophical Investigations* illustrates the difficulty of grounding images in meaning

2.1.2 Symbols

Symbols are amodal and do not necessarily bear any resemblance to the object they represent. As an analogy, take variables in algebra that stand for numbers and are manipulated the way numbers would be. Similarly, symbolic representations stand for objects in the world. However, because they are not inherently similar to the things, gross similarity cannot account for matching. Symbols are seminal to computationalism because they can be evoked as inputs and outputs of mental computations.

2.1.3 Feature spaces

A feature space is an N-dimensional space (N: number of features), in which vectors that represent individual objects are embedded. Many possible formulations of vectors exist, the simplest being binary arrays - arrays composed of 1s and 0s to indicate the presence or absence of features.
DiCarlo & Cox [15] presented a neurophysiological interpretation of feature spaces, wherein

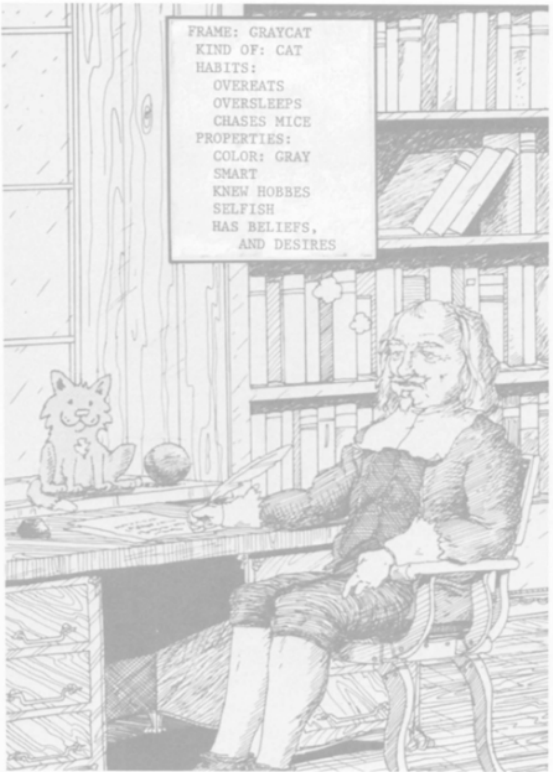


FIGURE 2.4: Hobbes uses semantic symbols to represent a cat with a ball. Source: Cummins (1991)[12]

the representation vector is a collection of visual neuronal responses embedded in a high-dimensional space. In addition to making manipulation straightforward, this formalism makes it possible to consider many types of features. There are two significant drawbacks associated with it.

1. Invariance

   To create a feature vector, one needs to identify features that match within a category but not across categories. These features should also be transformation-invariant. Hebart *et al.* [27] attempted to do this empirically through an odd-one-out task. Although they were able to extract feature dimensions that humans used for similarity judgments, the robustness of these dimensions is yet to be tested.

2. Curse of Dimensionality

   An increase in the number of dimensions causes an exponential increase in the number of examples required for learning. It is possible to reduce dimensions, but doing so while preserving invariant features is a challenge.

FIGURE 2.5: Hebb represents a cat with a ball as brain activity. Source: Cummins (1991)[12]

2.1.4 Structural Descriptions

Describing an object's shape or geometric structure in terms of generic components and spatial relationships forms its structural description. The parts and spatial relationships are standardized so that all object descriptions use the same vocabulary. The most well-known example of a structural description framework is Biederman's Recognition-by-Components theory [7]. It postulates a set of thirty or so primitive shapes called geons. Geons have a low likelihood of occurring in images by chance (*non-accidental*) and are well-suited to making inferences about 3D object structure.

Simple structural descriptions do not account for quantitative information present in images. While this makes them invariant to some extent, it also causes the loss of potentially relevant information (for instance, the relative sizes of parts). Descriptions based on geometric constraints address this shortcoming by maintaining a list of coordinates of prominent features. Alignment theories [55][41] use such descriptions to compute the hypothetical viewing positions of objects and re-align them for similarity computation. However, the identification and extraction of primitives in real-world images is a complicated and unreliable process (Figure 2.7). There is also degeneracy in structural descriptions (see Figure 2.8), meaning that multiple representations are possible for the same object. Additionally, comparing representations of this class is equivalent to labeled graph matching.

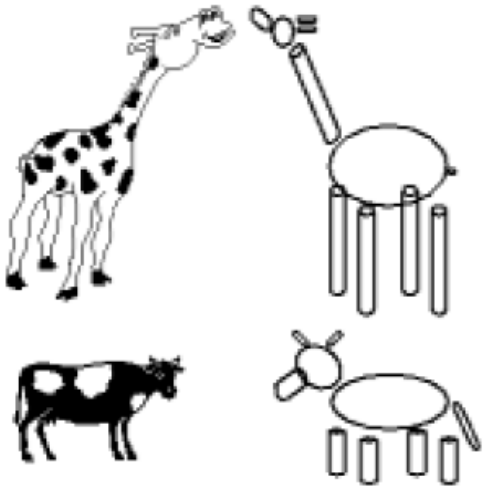


FIGURE 2.6: Structural descriptions employ primitive components to represent objects

Solving this problem is simple only when the descriptions are simple or when sub-optimal solutions are admissible.

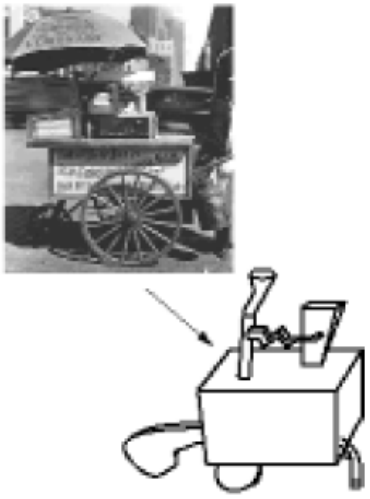


FIGURE 2.7: Real-world scenes, such as this NYC hot dog cart, are difficult to decompose into parts

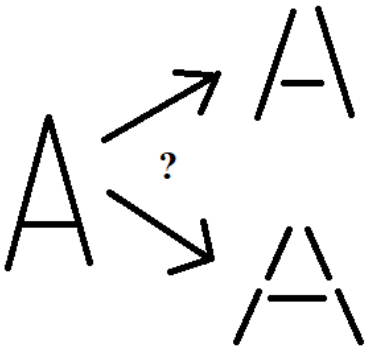


FIGURE 2.8: Degeneracy: The possibility of multiple structural descriptions for the same image

2.2 Conclusion

As one would expect, all significant theories associated with visual representations have drawbacks that cannot be ignored. In this work, we attempt to combine the generalizability and invariance of structural descriptions with the computability feature spaces. Instead of describing what objects look like, we define how they are drawn. These descriptions (which we call motor programs) capture the presence of relevant features in objects, are immune to transformations, and are easy to manipulate. Consequently, they allow object discrimination and can be used for object recognition with ease.

Chapter 3

Shapes, Skeletons, Drawings

The role of an object's shape in its recognition is intuitive to all those who can see. In chapter one, while trying to recognize tapirs, you likely compared the body structures of animals shown to you. It is infrequent for two similar-looking images to be structurally very different. Elder & Velisavljević [20] tested the significance of various cues in a rapid animal detection task and found that humans relied primarily on shape and texture for visual tasks. Carlson *et al.* [10] supported this with a MEG study, showing a negative correlation between the difference in shapes of two objects and how quickly the brain can tell them apart. An apposite shape dependency is also present in computer vision models. But outlines and contours are not the most dependable candidates for modelling object structure. In this chapter, we briefly review how the brain and machines represent geometric structures of objects, and lay the foundations for skeleton-based motor representations.

3.1 Shapes in the Brain

Inside the brain, shape processing is handled primarily by the ventral visual pathway [42]. Neuronal representations in the early stages, starting from the retina till V1 and V2, are confined to encoding *local* features. As the information proceeds towards V4 and the IT cortex, the receptive fields become larger and more complex, and representations are more *global*.

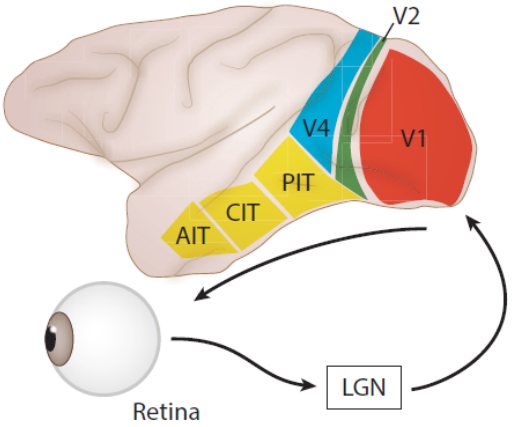


FIGURE 3.1: The anatomical path taken by visual information. Adapted from [16] and [19]

3.1.1 Areas V1, V2

In their famous experiment with cats, Hubel & Wiesel [28] demonstrated that neurons in Areas V1 and V2 (primary and secondary visual cortices) are tuned to local orientations. In comparison to V1, V2 receptive fields are larger and respond to more complex local features. However, representations in both areas comprise pointillist inputs from the LGN and are very similar.

3.1.2 V4 and IT Cortex

Area V4 is the gateway between early cortices and the inferotemporal (IT) cortex. The neurons here represent global contour information [9] and specific types of deformations [31].
There is empirical evidence that demonstrates invariant object recognition in the IT cortex [26] [46]. Interestingly, individual neurons in this area are not immune to variations in size, pose, position and background [14]. In fact, neurons that are more object-identity specific are not highly invariant[59] [16].

Despite extensive neurophysiological research, we do not fully understand how neuronal representations support visual perception. Though IT neurons seem to respond to object categories, how the brain compares what it sees with what it knows remains unclear. Evidence from psychophysics [40] [53] and physiology [44] indicates that cortical object

recognition is supported by view-based models, wherein collections of view-specific features constitute object representations. But contradicting data suggesting that the brain creates structural descriptions also exists, and consensus on the issue is wanting [51] [52].

3.2 Shapes in Computational Models

Computer vision models have improved exponentially and can now exhibit an almost human-like accuracy on recognition tasks. Studies with Convolutional and Deep Neural Networks show that they employ shape cues [36], attend first to the *bigger picture* (Global Advantage Effect) [30], and that their internal activity correlates well with IT responses [57][32] and BOLD activations [49]. At the same time, there exist significant differences between humans and these networks. For starters, their performance is sensitive to local features, unlike humans, whose similarity judgments depend primarily on global structures [5][6]. They are constantly outperformed by humans, and when they fail, they do so in ways that humans don't.

A large part of the computer vision research is devoted to modifying models to address these issues. But instead of dipping our feet further into the technicalities of computational models, let's turn our attention to the abstract shape representations inside of them.

In his review of computational models of object structure, Elder [19] distinguishes two classes of representations - generative and discriminative.

As the name suggests, generative representations capture the process of generation of the shape and are enough, in theory, to recreate it.

Discriminative representations describe how the shape is different from other shapes and have little to do with how it was created. Most modern models, including DNNs, employ representations of the latter kind. But they are only adept at discriminating classes from one another (cats vs dogs, for example) and not at describing images or computing similarity.

On the other hand, generative models represent structures in a way that is closer to our phenomenological experience of shape. Although they do not meet the performance benchmark set by DNNs, recent evidence suggests that 3D generative models can explain behavioural responses better than their discriminative counterparts [22].

While an object's shape is seminal to both its neuronal and computational representations, an outline or contour-based description of structure is not without impediments. For instance, exemplars of the same category have different outlines as illustrated in Figure 3.2. Contours are also sensitive to transformations and minor deformations along edges (Figure

3.4). Due to these issues, representations composed of contours or outlines are unstable and difficult to learn.

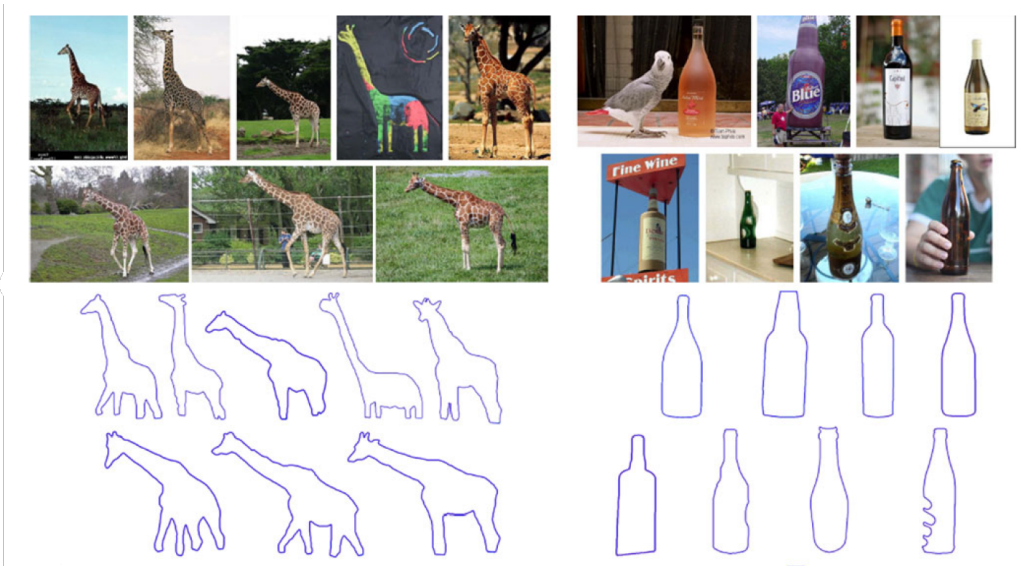


FIGURE 3.2: Images of giraffes (left) and bottles (right), and their corresponding outlines (adapted from [54])

FIGURE 3.3: Contours are sensitive to perturbations [4]

To make the representations stable, object structure needs to be described in a way that would capture the global shape while being immune to transformations. Shape skeletons are a class of models capable of doing this.

3.3 Shape Skeletons

Shape skeletons represent a shape through its medial axis - the set of centers of circles that fit into the shape maximally [25]. Medial axis is also defined as the set of points in a shape equidistant from the shape boundary at least two boundary points [11]. Medial

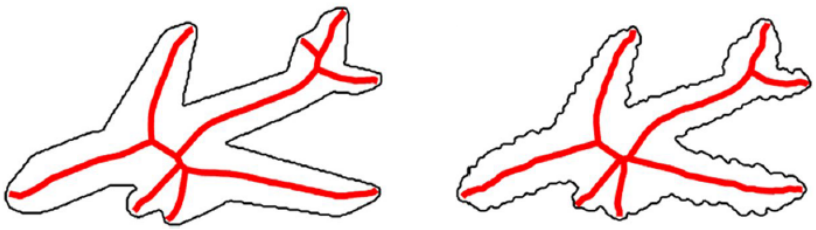


FIGURE 3.4: Medial axis skeletons are immune to perturbations [4]

axis-based skeletons are compact and low-dimensional, making them easy to learn, and consequently, suitable for few-shot learning.

Experimental research also supports the prominence of skeletal structures in perception. Recordings from the early visual cortex [38] reveal that it computes and represents medial axes of shapes, and that it is possible to decode skeletons from IT cortex activity [29]. When looking at Gabor patches, human subjects display greater contrast sensitivity when the patches are close to the shape's medial axis [33]. Additionally, they are able to recognize images despite changes in components as long as the medial axes stay the same. Shape skeletons have been used extensively in computer vision research as well [39][48]. Human performance on superordinate classification tasks is well-explained by Bayesian classifiers that use skeleton parameters [56]. Trinh and Kimia's skeleton-based model [54] performs segmentation and classification tasks accurately and is invariant to simple transformations. But modern recognition models rarely use skeleton information, relying instead on feature extraction [35]. This non-usage does not seem detrimental to their performance - as we saw in section 3.2, they closely match behavioral and neural responses. Ayzenberg & Lourenco [4] addressed this dichotomy and compared a model of skeletal similarity with the Gabor-Jet model, GIST, HMAX, and AlexNet, finding the skeletal model to be most predictive of human judgments.

3.3.1 Pruned Medial Axis Models

A typical medial axis structure, called the medial axis transform (MAT), is hierarchical, composed of a parent branch and several secondary branches growing off it. The parent captures the shape's global geometry, and its children describe local variations. MAT, just like contours, is sensitive to subtle shape changes. Even a minor alteration can cause the formation of additional secondary branches in an MAT skeleton (Figure 3.5).

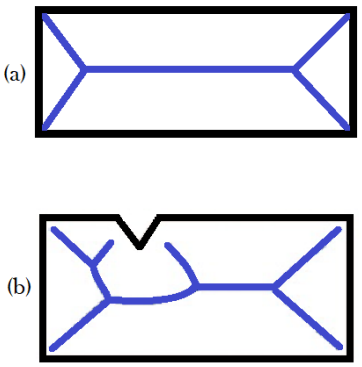


FIGURE 3.5: (a) A rectangle and its medial axis skeleton (in blue).
(b) A nook in the rectangle causes the formation of secondary branches
Adapted from [23]

In an alternative known as the Pruned Medial Axis Model, the skeleton is made simpler by *pruning away* some branches and preserving only those that describe the object's overall shape. As expected, pruned models are more stable across perturbations than non-pruned ones. Human responses on behavioral tasks are best fit by models that prune the medial axes in some way [3] [33]. Therefore, in our work, we employ simplified shape skeletons to represent natural objects and expect that skeletal representations will be invariant to rotation, scaling, and translation.

Chapter 4

Bayesian Program Learning

In the previous chapter, we brought up three significant drawbacks of artificial recognition systems - they are unable to generalize without extensive training, they are outperformed by humans, and they fail gracelessly. We also claimed (in Chapter 1) that one of the primary reasons for these drawbacks is that visual features alone are not discriminative enough. Further, we posited a new feature – motor programs (descriptions of how objects are drawn). In this chapter, we present the Bayesian Program Learning framework [37], which we used to infer motor programs given images.

Programs are collections of steps that can be combined recursively to generate concepts [1]. In Bayesian Program Learning, programs are probabilistic and compositional sequences that produce drawings of visual concepts. Their probabilistic nature and compositionality allow them to learn from limited data, generalize their knowledge to unseen examples, and possess a human-like inductive bias. It is worth mentioning here that the original BPL framework was trained on the Omniglot dataset (Figure 4.2) that contained drawings of letters from different scripts. However, the context in which we deploy it in our work (i.e., drawing medial axis skeletons of natural images) is significantly different. This difference is a possible reason for BPL's inability to create unique motor representations for our images (discussed further in Chapter 6). In the following sections, we briefly describe BPL's components and present it as a robust candidate for few-shot representational learning.

[1] See [21] for a more formal treatment of programs

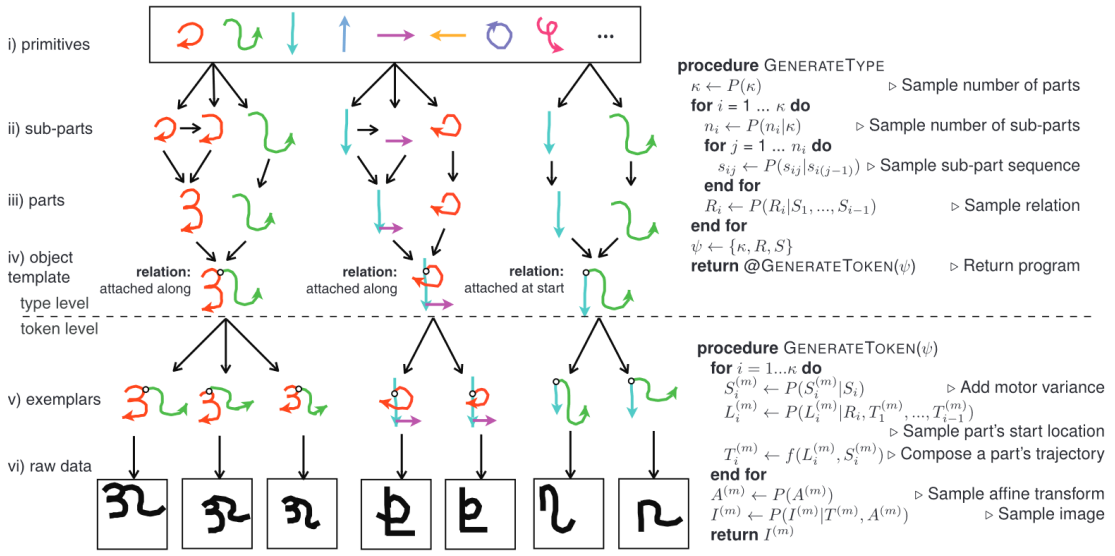


FIGURE 4.1: Generating types and tokens of concepts from primitives. The algorithm samples numbers of parts and subparts, and combines sub-part sequences according to spatial relations to form types. Motor variance and start location variability are introduced to form tokens of types. Source: Lake, Salakhutdinov, Tenenbaum (2015). [37]

4.1 Bayesian Program Learning

The BPL framework, at its core, is composed of learned primitives. Primitives form *sub-parts*, which, in turn, combine to form *parts*. Parts are joined together according to *spatial relations* to create *types* of concepts (such as A, B, ball, cat). Each concept type is represented as a lower-level generative model. The lower-level model employs noise and motor variance to create many exemplars (called *tokens*) of each type.

4.1.1 Training

The Omniglot dataset contains handwritten characters from 50 scripts that were split in a 3:2 ratio for training and evaluation, respectively. Hyperparameters for primitives, start positions, spatial relations token variability, and image rendering were learned during training.

As mentioned previously, primitives are the atoms of BPL. They are simple curves that can be combined to form bigger and more complex entities. Each primitive has five unique features - an identification index z, hyperparameters of a Gaussian distribution (μ, σ; used while generating character types), and hyperparameters of a Gamma distribution (α, β; also used while generating character types).

FIGURE 4.2: The Omniglot dataset consists of handwritten characters from various scripts

In the dataset, each drawing consisted of strokes (sequences between the pressing down and lifting up of the pen) that were split further into sub-strokes (sequences separated by short pauses of the pen). Sub-strokes were normalized in time and space, fit with a spline, and represented by five control points (in $\mathbb{R}^{10}$) each. After using a diagonal Gaussian Mixture Model to partition sub-strokes into primitive elements, hyperparameters $(z, \mu, \sigma, \alpha, \beta)$ for each primitive were inferred using Maximum Likelihood Estimation.
The image plane was discretized and a multinomial grid model was fit to learn stroke start positions. Four kinds of spatial relations (independent, along, start, end) were learnt by assuming temporary values of parameters and refitting the model. Finally, to learn image parameters, the centre of mass and range of inked pixels was computed. Each character image was transformed such that its mean and range matched the group average of all images for that character. Maximum Likelihood Estimation was used to estimate the ink hyperparameters (a, b) for image drawing.

4.1.2 Motor Program Inference

BPL follows a bottom-up inference method to produce a large set of possible motor programs, which can be refined using optimization and MCMC.

4.1.2.1 Image Thinning

Inference begins with thinning the input image to 1 pixel width. The lines and forking points of the thinned image act as edges and nodes of an undirected graph.

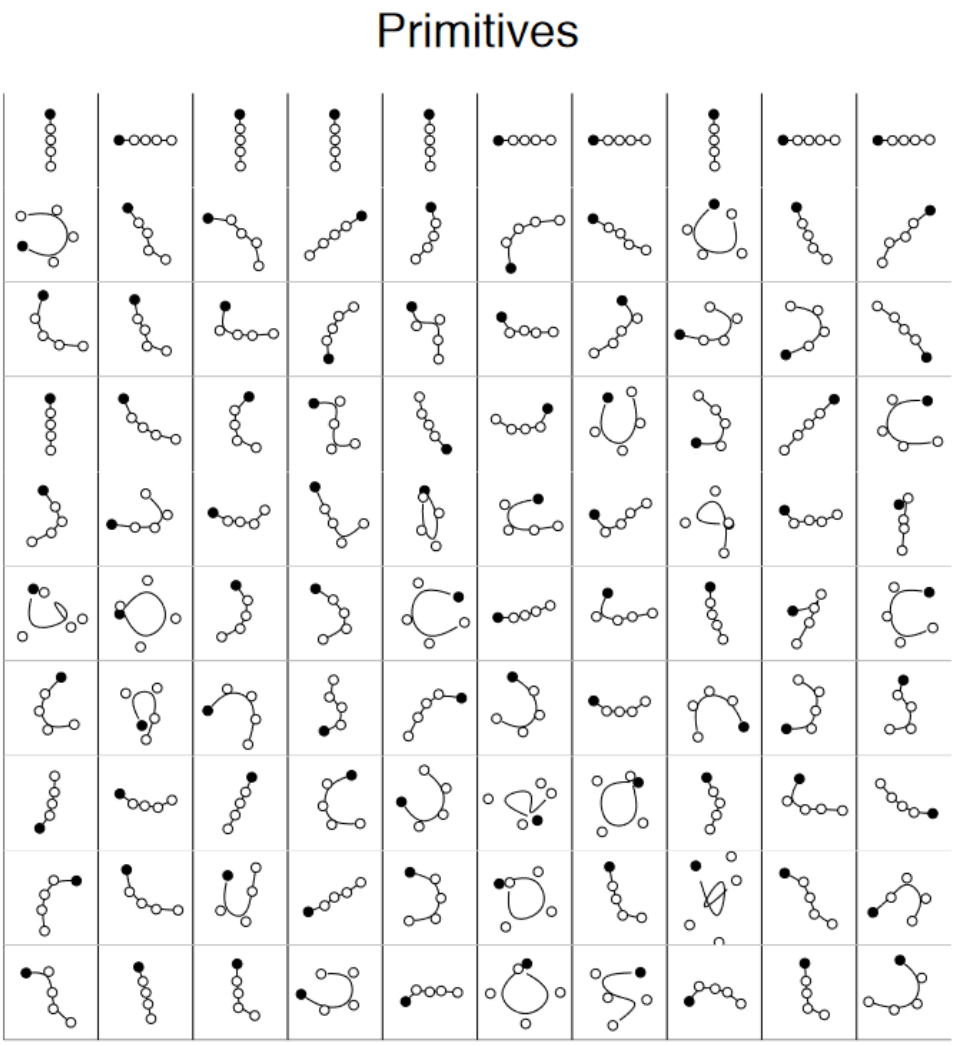


FIGURE 4.3: Primitives extracted during training the BPL model. The circles represent control points. The first control point is filled with black. Source: Lake, Salakhutdinov, Tenenbaum (2015). [37]

4.1.2.2 Random Walk

Each parse of the input image is created by taking a random walk over it until all the edges have been traversed at least once. The number of parses that can be generated in this manner grows exponentially with the number of edges, so the random walks are made to be biased. During a walk, the probability of an action A being chosen is proportional to the local angle θ_A around the stroke such that actions that minimize θ_A are preferred (Figure 4.5).

$$P(A) \propto \exp(-\lambda\theta_A) \tag{4.1}$$

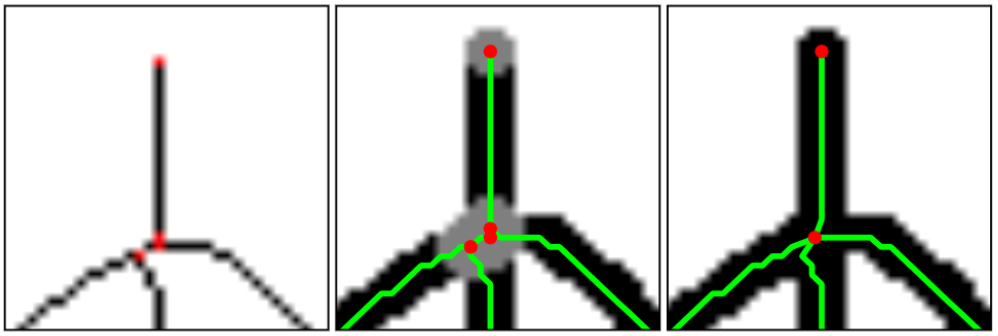


FIGURE 4.4: (Left) An image thinned to 1 pixel width. (Middle) Imperfection in detection of forking points can lead to multiple points in close vicinity. Maximum circle criterion (illustrated as a shaded circle here) is applied to forks to correct this. (Right) Forking points are merged to produce the final thinned image. Source: Lake, Salakhutdinov, Tenenbaum (2015). [37]

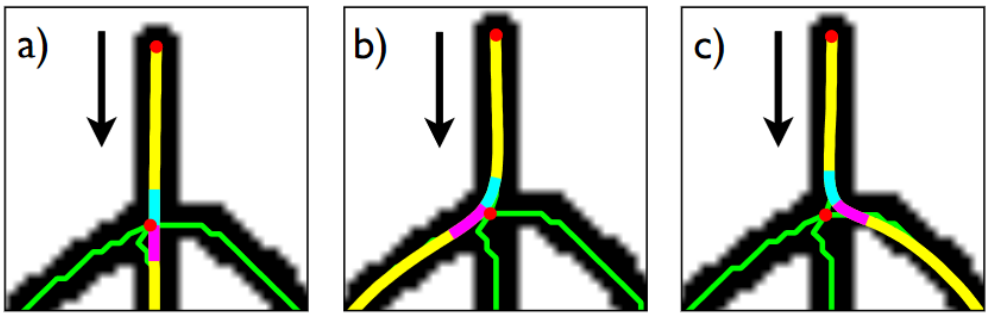


FIGURE 4.5: When the random walk proceeds from top in the direction of the arrow, it can proceed in three ways: In (a), the local angle is zero degrees. In (b), it is 28 degrees, and in (c), it is 47 degrees. Source: Lake, Salakhutdinov, Tenenbaum (2015)[37].

Once the parse has been created, the strokes in it are smoothed and divided into sub-strokes using a greedy search. The substrokes are classified as primitives z_i and the decomposition is scored by a generative model for strokes. The best K models are then optimized, fine-tuned, and returned as output.

Chapter 5

Experiments and Results

So far, we have seen how machines and humans are different and how current Machine Learning techniques of few-shot learning (such as data augmentation) are centered around creating more data from existing data points, and not on the learning process itself. We believe that this is the reason for the graceless failure [1] of recognition algorithms.
For improved learning of representations, we presented shape skeletons as the ideal, generalizable feature. Skeletons capture the overall geometry of shapes and are not affected by local contour changes, 2D rotations, or scaling. But comparing skeletal similarity in the visual space leaves room for error and inefficiency. We also proposed that skeletons should not be compared on the basis of *how they look*, but *how they are drawn*. Our expectation in doing so is that higher motor similarity correlates with higher perceived visual similarity. However, motor similarity metrics for geometric structures are non-existent in the current literature. In the following sections, we present and summarize the metrics we defined and the analyses we performed using them.

5.1 Data

We prepared a dataset consisting of 50 natural object categories. We used CIFAR-10[34], CIFAR-100[34] and ImageNet[13] to obtain category labels (Appendix A) and sourced exemplar images from the internet.
For each exemplar image, we created 10 transforms - 4 rotational (rotated in 2D), 4 scalar, 2 combined (both rotational and scalar)(see Figure A.2). Then, we applied a medial axis transform to each transformed image to obtain its skeleton. 13 categories were excluded

from further analysis due to the skeletons being too simple (straight lines) or too complex. In total, we had 10 skeletons per category, resulting in a cumulative of 370 skeletons.

5.2 Analyses

We used the BPL framework to infer motor programs for skeletons and obtained stroke-wise coordinates as outputs. As mentioned earlier, the primary challenge was that of representing these visual drawings in motor space. Our first approach involved capturing the drawing process in the form of motor program IDs.

5.2.1 Program ID

We identified and indexed six functions in the BPL framework that, in our opinion, formed the core of the drawing process. These functions were:

1. *pts_on_new_edges*: Make a list of points that can act as potential start locations for drawing
2. *pen_up_down*: Put the pen down at an unvisited edge
3. *pen_simple_step*: Make a simple move
4. *pen_angle_step*: Select a move based on angle from the current trajectory
5. *angles_for_moves*: Compute the angle between current trajectory and each possible move
6. *action_via_angle*: Pick the next move depending on the angle computed. Move probability proportional to $e^{\frac{-\lambda\theta}{180}}$.

The idea was that these functions would represent not the drawing, but how it had been created (Figure 5.1). We stored the order in which these functions had been called (and the angle chosen by BPL) and put them together to form motor IDs for all the skeletons. Some exemplar motor IDs are as follows:

- 21345[1.2074e-06]621345[45]645[45]6213
- 21345[29.7449]645[45]645[40.6013]621345[29.7449]645[45]621321345[29.7449] 621345[40.6013]6

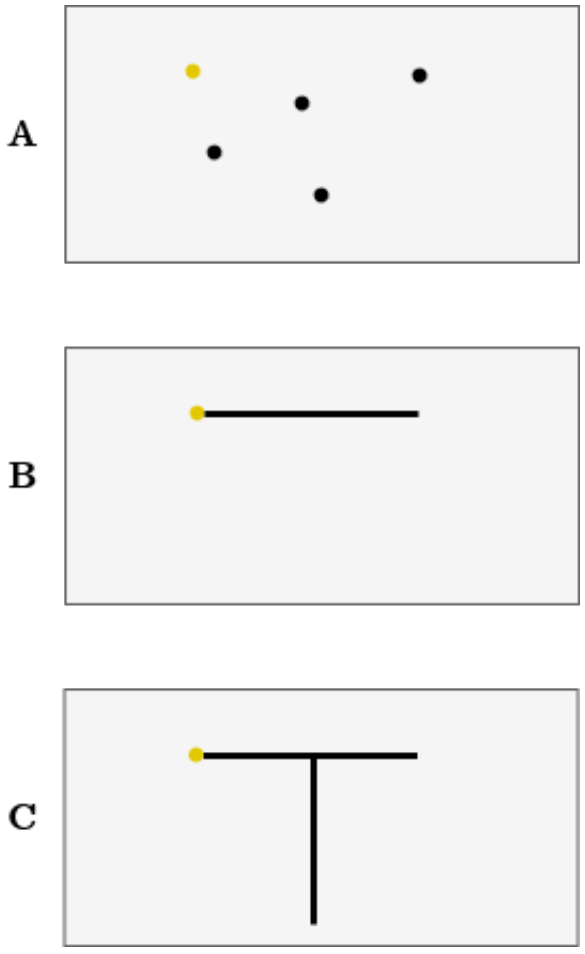


FIGURE 5.1: Drawing the letter 'T'. (A) The model samples start points for a stroke (in black) and selects one randomly (in yellow). (B) The pen is put down and a move is made. (C) The model compares all possible successive moves and selects the most probable move. The process is repeated till the completion of drawing.

- 21345[7.1250]645[0]645[45]645[45]621345[45]621321345[45]621345[45]645[45]621345[0]621345[45]621345[0]645[7.1250]645[45]621321345[45]645[45]621345[0]645[45]6213

Once represented in this manner, motor programs could be compared easily through their IDs. Similar skeletons would be drawn in a similar way, leading to identical IDs. To test this, we calculated the similarity distance between each pair of IDs.

$$\text{Similarity} = lev + ang \tag{5.1}$$

- The Levenshtein distance between two strings is the minimum number of single-character mutations (insertions, deletions or substitutions) required to make them

identical[43]. For two strings a and b,

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_i)} \end{cases} & otherwise \end{cases} \tag{5.2}$$

where i and j are the terminal character positions of a and b, respectively.

- The angle distance is the Euclidean distance between two corresponding angles.

$$ang = \frac{\Sigma(angle1 - angle2)}{min(n2, n1)} \tag{5.3}$$

Figure 5.3 shows a 370 x 370 matrix where each cell represents the distance between two IDs. Each row displays how similar a single skeleton is to all others. We expected intra-category distances to be smaller than inter-category distances. That is, distances along the diagonal should have been lesser than distances farther away from the diagonal. But, as can be seen, this was not the case.

5.2.1.1 Discriminability of Program IDs

To test whether program IDs were discriminable at all, we reverse-engineered drawings from IDs (Figure 5.4). Note that the drawings were not recreations of the original skeletons, but we expected them to be visually distinguishable. We tested discriminability using a pre-trained nearest neighbours model. For each category, we chose a transformed image of the object and used the model to compute similarity distance between our chosen image and other images of the same category (*Within-Category Image Distance).* We also computed similarity between the image's skeleton and other skeletons of the same category (*Within-Category Skeleton Distance*), and between the image's ID drawing and other drawings of the same category (*Within-Category Drawing Distance*). Figure 5.5 shows the distances for an exemplar category. We repeated this process for between-category comparisons. Figure 5.6 shows Between-category Image, Skeleton and Drawing distances. On an average, for within-category comparisons, distances between skeletons matched distances between the original images more closely than the distances between ID drawings. Between-category image similarity did not correlate reliably with skeletons or drawings.

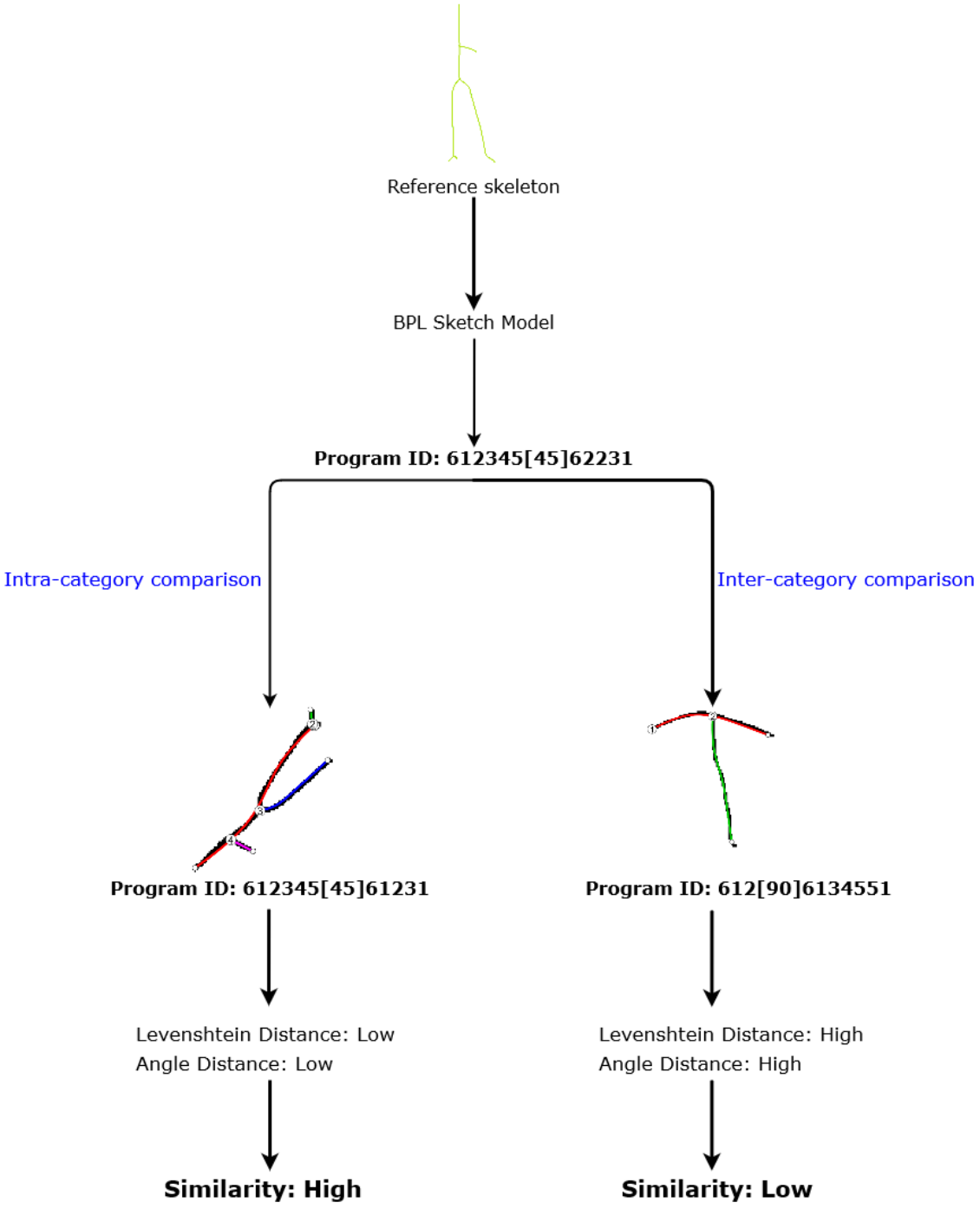


FIGURE 5.2: Computing similarity between Program IDs

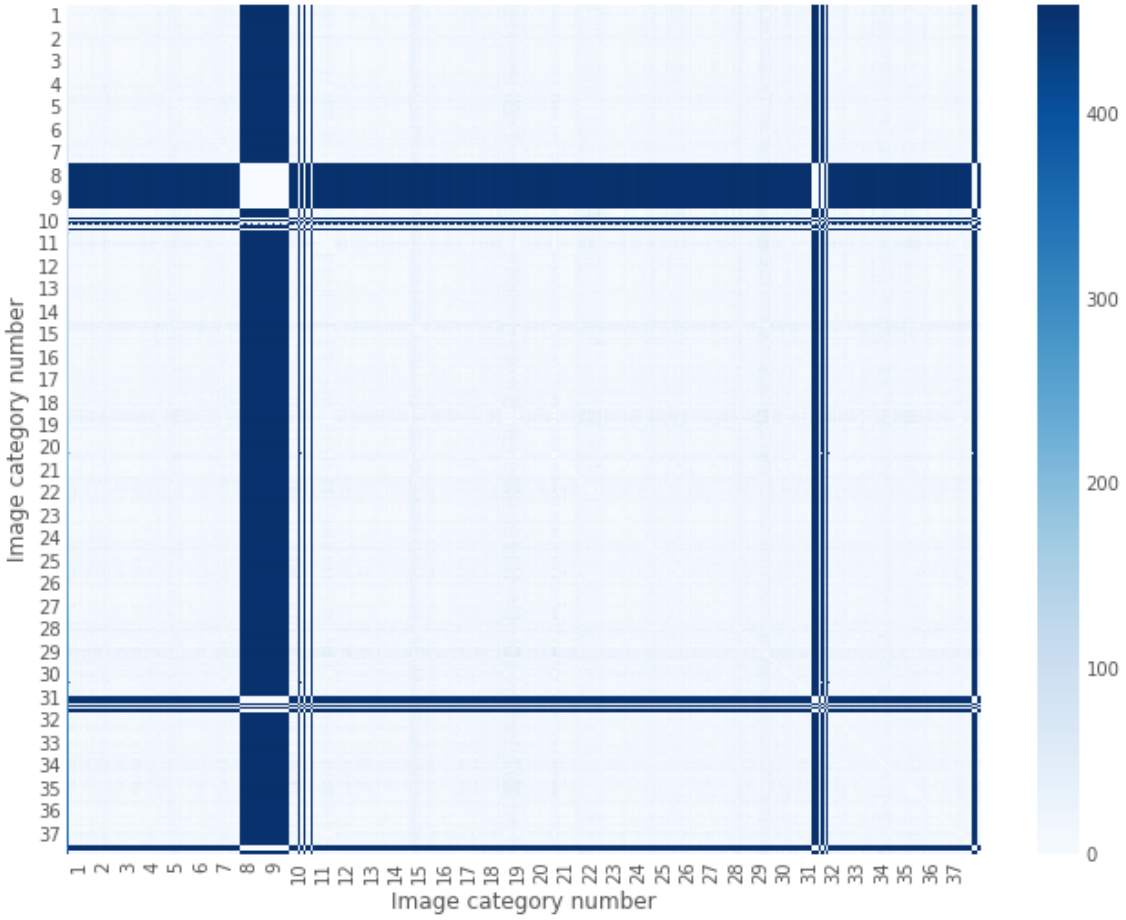


FIGURE 5.3: Program ID similarity matrix for 37 categories. Each row shows the Levenshtein distance between a single skeleton and 370 others.

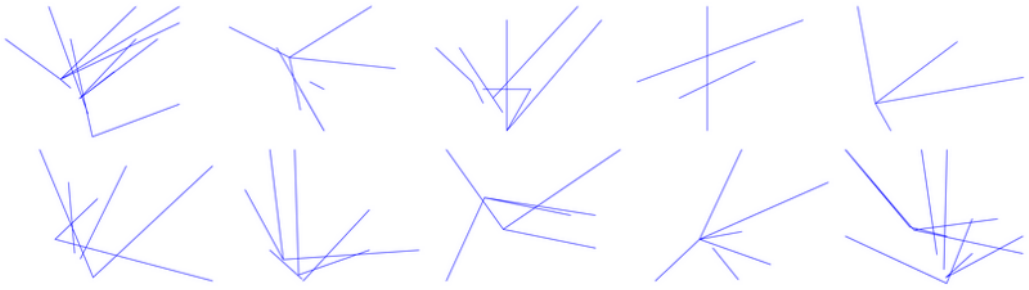


FIGURE 5.4: Drawings created from motor program IDs

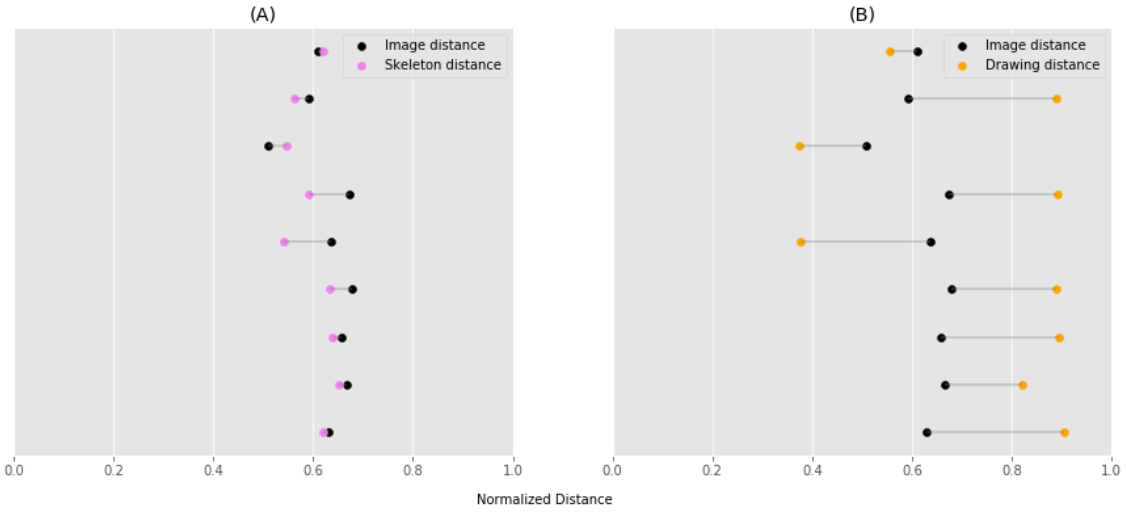


FIGURE 5.5: Within-Category visual distance between (A) images and skeletons, and (B) images and drawings. Image-Skeleton distances smaller than Image-Drawing distances.

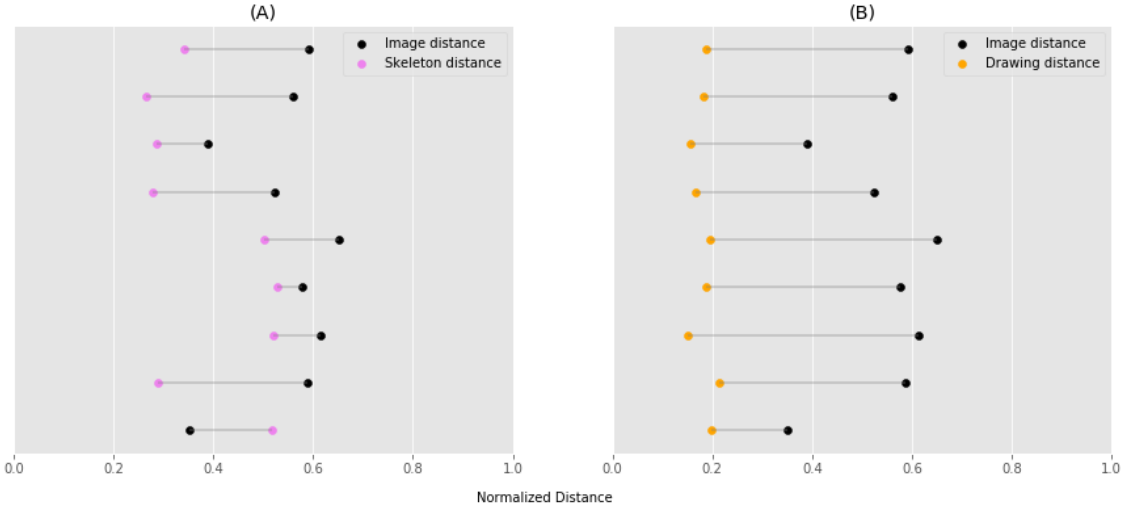


FIGURE 5.6: Between-category visual distance between (A) images and skeletons, and (B) images and drawings.

Hence, we concluded that program IDs were not informative enough to distinguish between objects within and across categories.

5.2.2 Grid Representations

In our second attempt, we used the stroke-wise coordinates provided by BPL to create invariant numerical representations. We visualized the drawing space as a 5 X 5 grid and

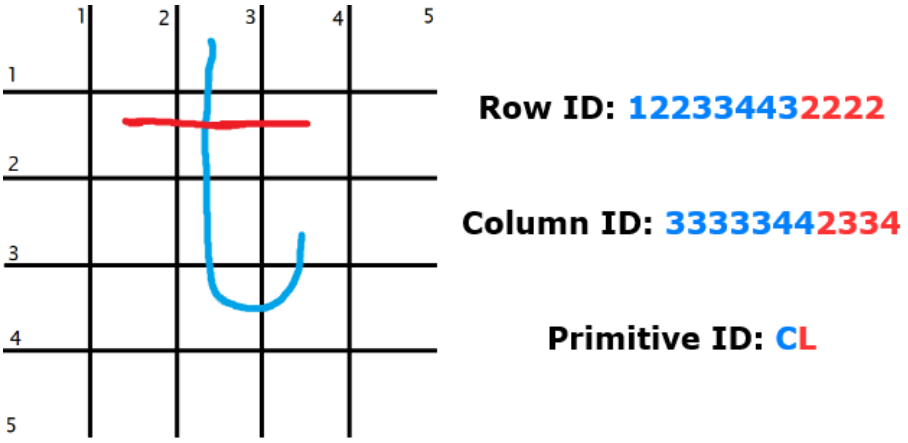


FIGURE 5.7: Drawing space divided into a 5 x 5 grid. Stroke 1 illustrated in blue and Stroke 2 in red.

represented strokes vis-a-vis their location on the grid. We did this as follows (See Fig 5.7 for an example):

1. **Row ID**: We indexed the rows on the grid from 1 to 5 and for each stroke, we replaced each y-coordinate with the index of the row it lay in.
2. **Column ID**: We indexed the columns from 1 to 5 and for each stroke, we replaced each x-coordinate with the index of the row it lay in.
3. **Primitive ID**: We classified each stroke as a straight line (denoted by 'L') or a curve (denoted by 'C').

We calculated Levenshtein distance separately for the row, column and primitive IDs, and computed the overall similarity as the mean of the three (see Figure 5.8 for example). Once again, we expected a significant difference between the intra and inter-category Levenshtein distances. The results looked more promising than the last time - we saw low distance throughout the diagonal. However, the distances did not seem to increase as we moved away from the diagonal. We took each image's 9 nearest neighbours and counted how many belonged to the same category as the image[1]. We found that for most images, a larger proportion of neighbours were from other categories 5.10. Therefore, we concluded that grid representations are not capable of object category discrimination, either.

[1]Because we were working with 10 images per category, in an ideal scenario, all of the 9 nearest neighbours would have been from the same category

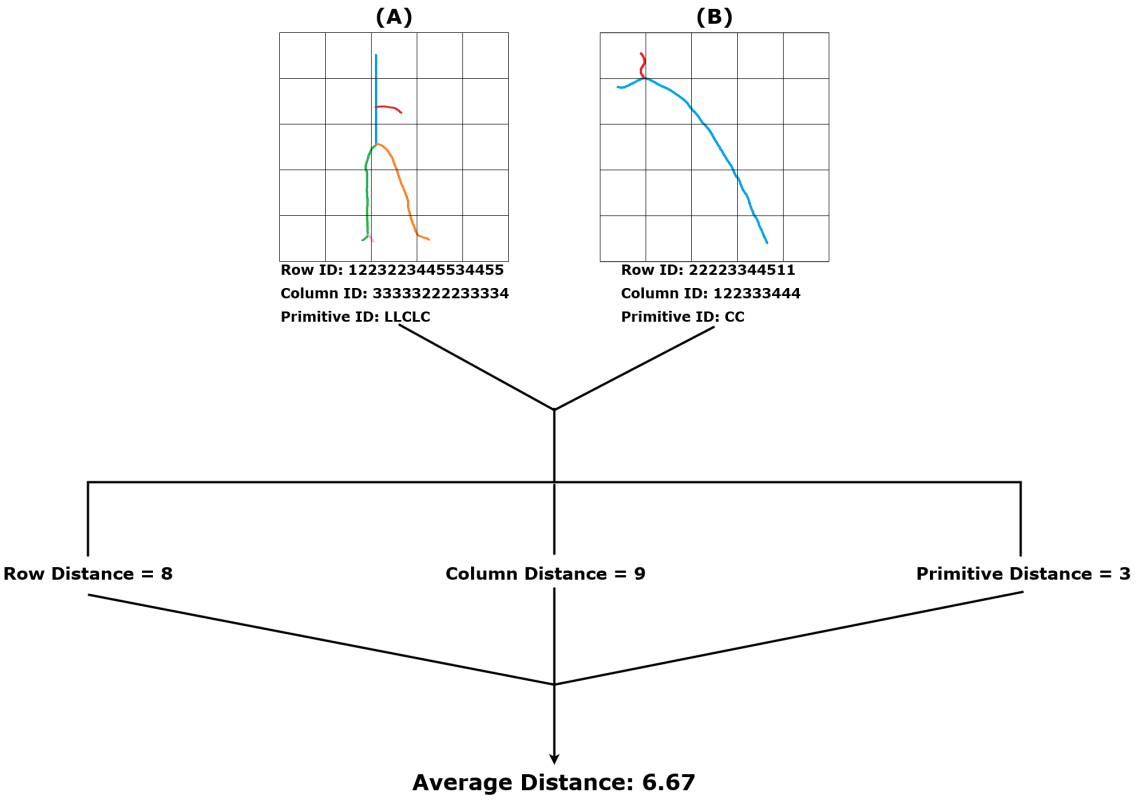


FIGURE 5.8: Grid Similarity: We computed the Levenshtein Distance between the Row, Column and Primitive IDs, and averaged the three to obtain the distance between skeletons.

5.2.3 Adjacency Matrices

Instead of capturing the process of drawing skeletons, we sought to describe the skeletal structure itself. To make these descriptions invariant, it was imperative to find a representation system that did not rely on coordinates.

An adjacency matrix is a square matrix, wherein the elements indicate whether two vertices are connected to each other. For each skeleton in our dataset, we used BPL to represent connections between nodes in the form of undirected adjacency matrices and calculated dot products between all pairs of matrices.

Transforming images does not change the number of nodes and their placement in the skeletons. It is, however, possible for nodes to get interchanged (as in Figure 5.11), which further causes rows and columns of the matrix to change. To account for that, we performed row and column permutations on the matrices before calculating the dot product.

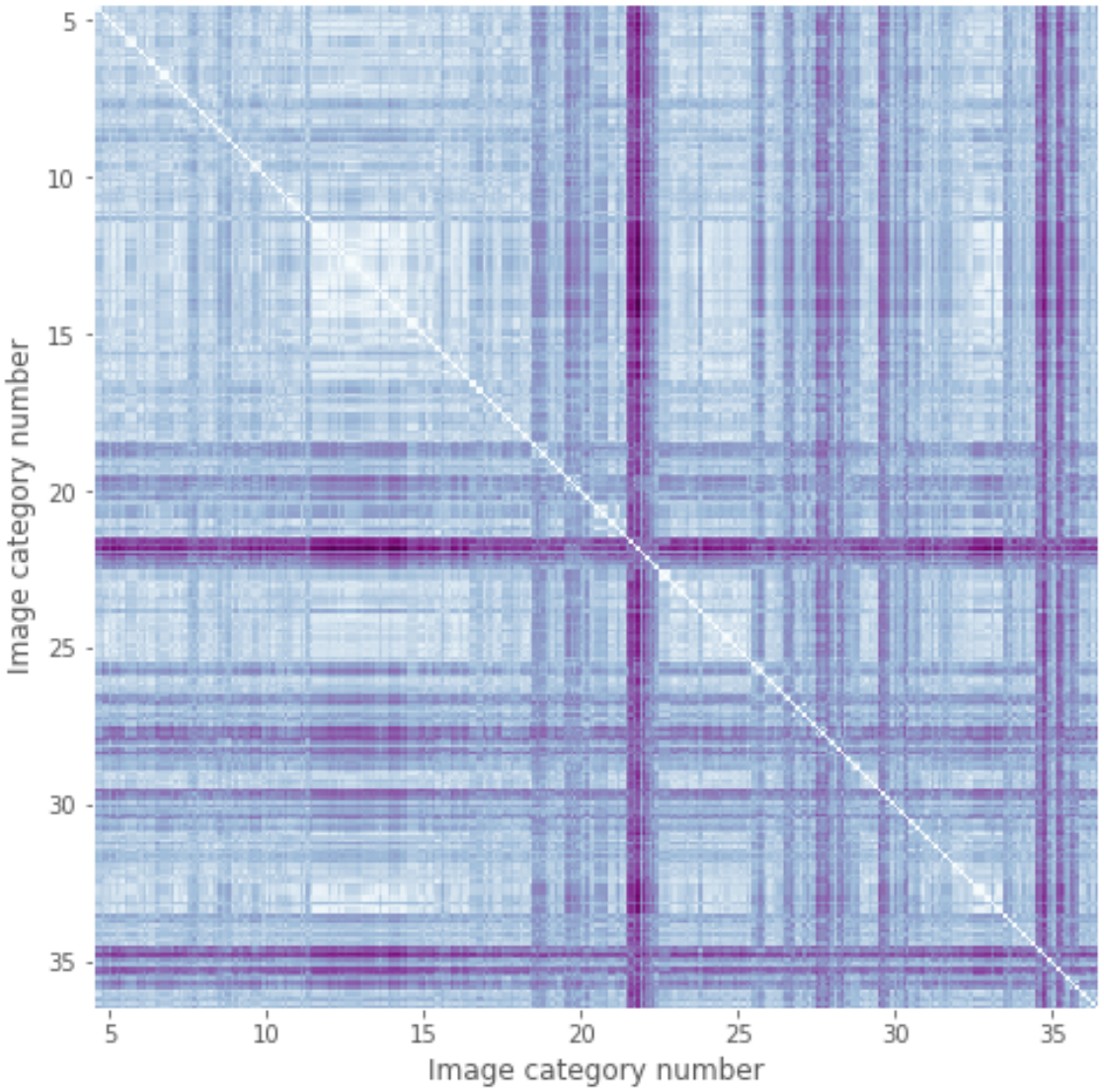


FIGURE 5.9: Distances between grid representations of image skeletons

Overall, the process was as follows:
For each pair of matrices (called M_1 and M_2)

1. Check if both are equal in size
   (a) If they are, proceed.
   (b) Else, let B and S ($B, S \in [M_1, M_2]$) be the bigger and smaller matrices, respectively.
   Let max_product = 0 and $N = Length(S)$. For each N X N subset of B:
      i. product = $subset \cdot S$

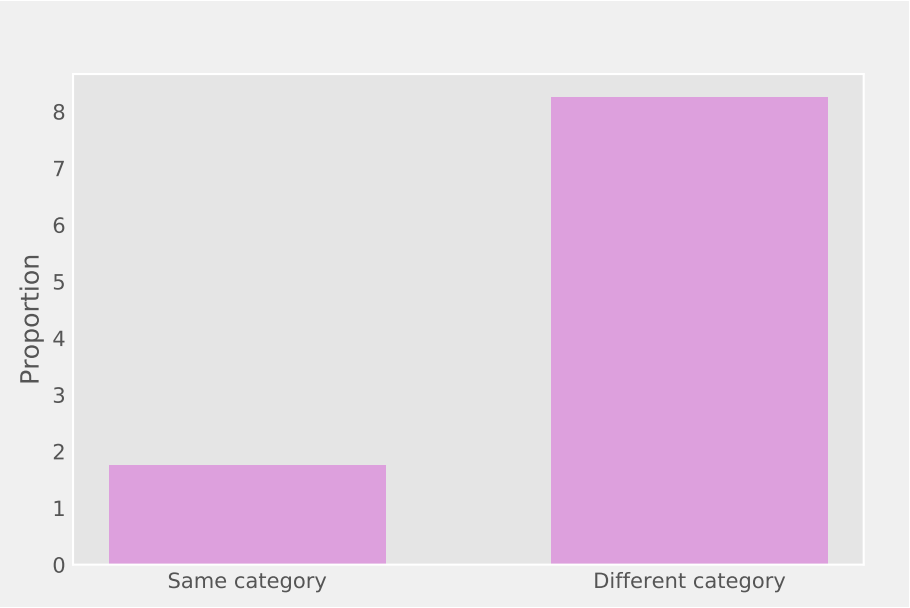


FIGURE 5.10: Proportion of nearest neighbours that belonged to the same category as the reference image was lower than those that belonged to a different category

ii. If product > max_product, max_product = product and $B = subset$

iii. Else, proceed

2. Normalization Factor $NF = Sum(M_1) + Sum(M_2)$

3. For P_i in $P = [P_1, P_2, ...]$ where P: set of all possible permutations of M_2, compute

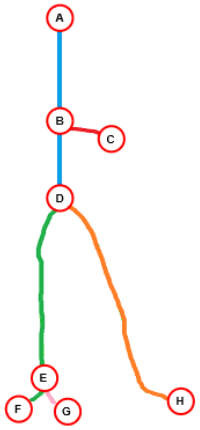
$$\frac{M_1 \cdot P_i}{NF}$$

4. Store the largest normalized dot product

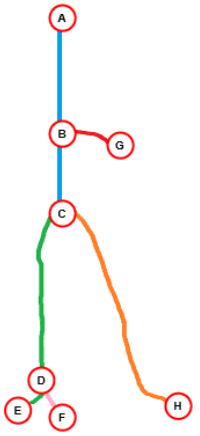
As a first-pass analysis, we repeated this process for 4 categories. Dot products of adjacency matrices are illustrated in Figure 5.12. The algorithm involved computing $N * N$ permutations for each matrix with N rows and N columns. As N crossed 8, permuting rows and columns turned into a computationally expensive process.

5.2.3.1 Depth-First Search

To reduce the size of the permutation search space, we introduced a Depth-First Search mechanism into the algorithm.
For each pair of matrices M_1 and M_2:



	A	B	C	D	E	F	G	H
A	0	1	0	0	0	0	0	0
B	1	0	1	1	0	0	0	0
C	0	1	0	0	0	0	0	0
D	0	1	0	0	1	0	0	1
E	0	0	0	1	0	1	1	0
F	0	0	0	0	1	0	0	0
G	0	0	0	0	1	0	0	0
H	0	0	0	1	0	0	0	0



	A	B	C	D	E	F	G	H
A	0	1	0	0	0	0	0	0
B	1	0	1	0	0	0	1	0
C	0	1	0	1	0	0	0	1
D	0	0	1	0	1	1	0	0
E	0	0	0	1	0	0	0	0
F	0	0	0	1	0	0	0	0
G	0	1	0	0	0	0	0	0
H	0	0	1	0	0	0	0	0

FIGURE 5.11: (Top) Sample adjacency matrix. (Bottom) Interchanging nodes changes the matrix.

1. Adjust size like before
2. Define a threshold value, $T \in [0, 1]$
3. Let P be the set of possible permutations of M_2. $P = [P_1, P_2, ...]$. For each P_i:
   (a) $D_i = M_1 \cdot P_i$
   (b) If $D_i \geq T$
      i. Find rows and columns which are identical between M_1 and P_i
      ii. Keeping the identical rows and columns fixed, continue to permute the non-identical rows and columns
      iii. Compute dot product for each permutation

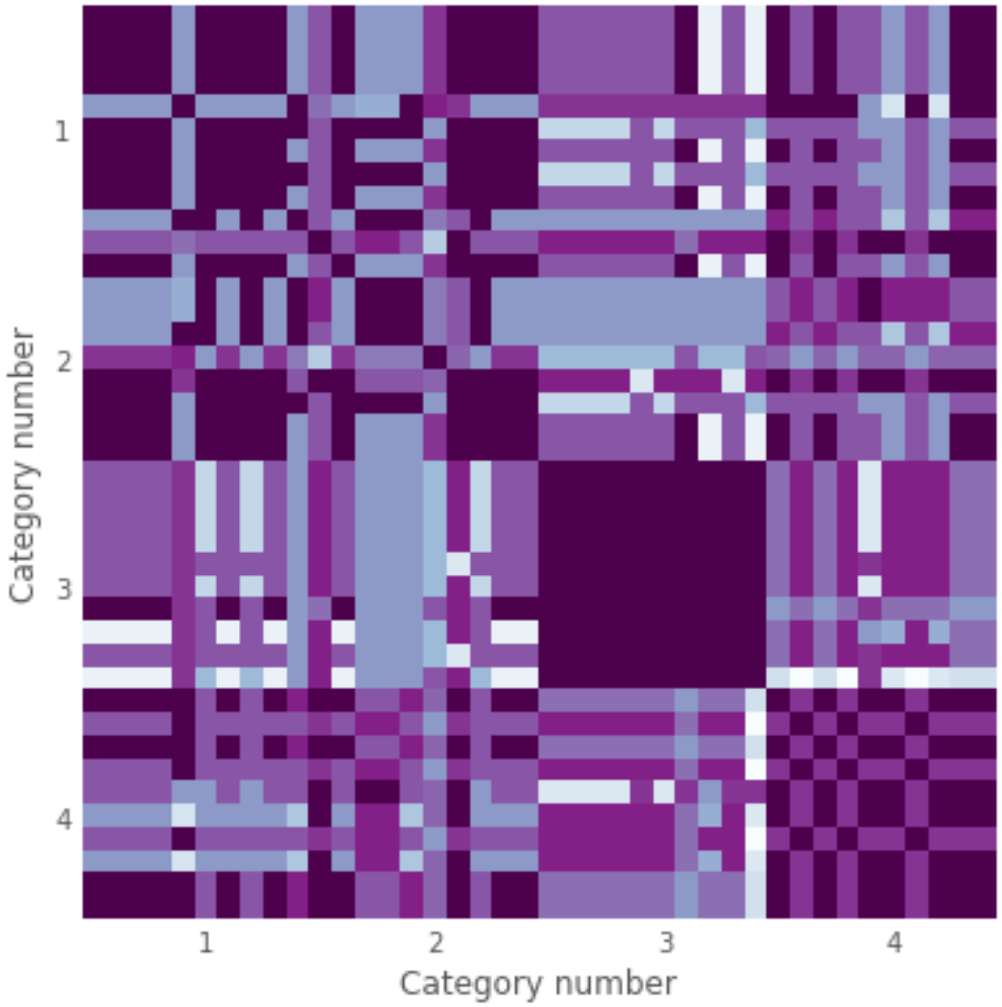


FIGURE 5.12: Adjacency matrix similarity. Values along the diagonal represent intra-category similarity, while those off the diagonal represent inter-category similarity. Intra-category similarity appears to be higher than inter-category similarity in Category 3, but this is not the case for Categories 1 and 2.

iv. D = largest dot product

v. Break

(c) Else if $D_i < T$, store D_i and move to next P_i

4. $D = max([D_1, D_2, ...])$

With a threshold value of 0.8, we were able to reduce the computational complexity to some extent. Figure 5.13 shows the dot products obtained through our DFS algorithm as a similarity matrix. The difference between intra-category and inter-category dot products did not appear to be significant, leading to the conclusion that adjacency matrices alone were not useful for category discrimination.

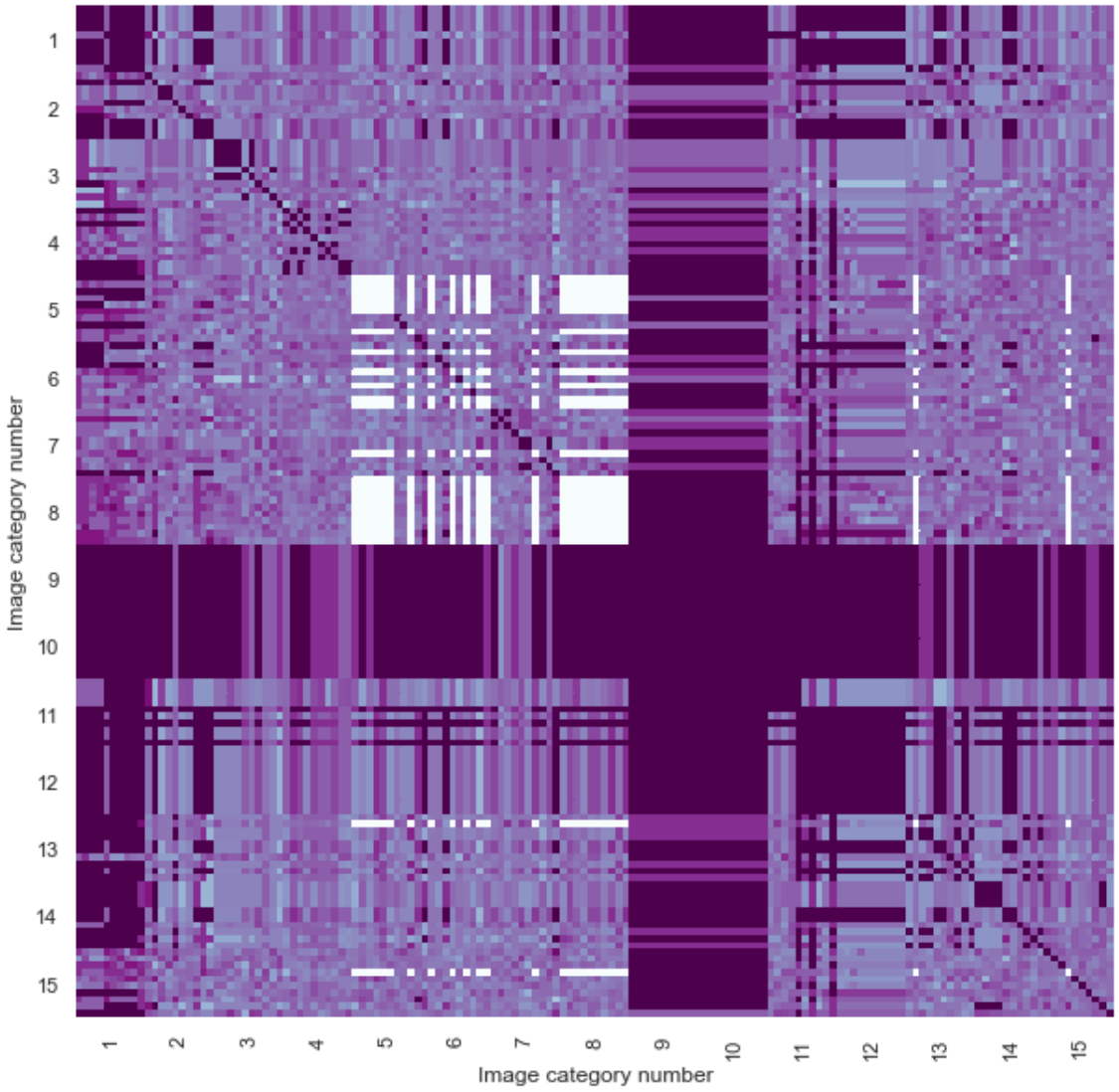


FIGURE 5.13: Adjacency matrix similarity computed after Depth-First Search

5.2.4 Adjacency Matrices and Edge Lengths

To make matrices more informative, we added relative stroke length information to the representations. For each matrix, we normalized stroke lengths and rearranged node indices according to decreasing length order (Figure 5.14). After the rearrangement of nodes (and consequently, the adjacency matrices), we computed dot products as before. We did not observe any significant improvements over previous analyses. Next, we did a comparative reassignment of node identities. Instead of rearranging matrix nodes independently for each matrix, we did the following. For a reference image I_1 (matrix M_1) and test image I_2 (matrix M_2):

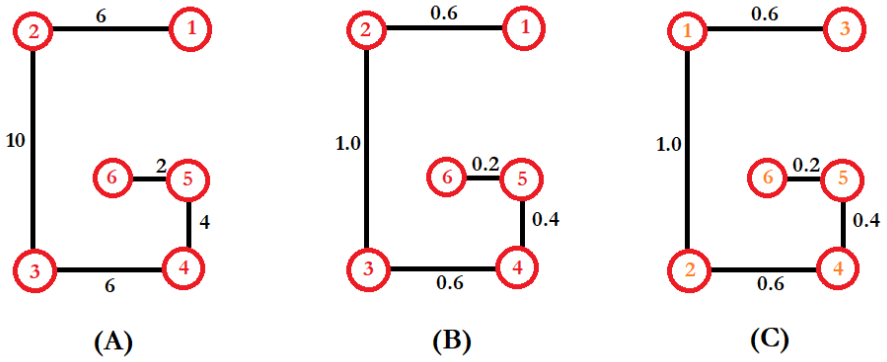


FIGURE 5.14: (A) Reference image. Lengths are displayed adjacent to edges, node indices are displayed in red. (B) Normalized stroke lengths. (C) Nodes rearranged in order of decreasing length.

1. Normalize stroke lengths in I_1 and I_2
2. Let T_1 and T_2 be the terminal nodes in M_1. Rename the terminal nodes in M_2 to T_1 and T_2.
3. Let L_1 and L_2 be the nodes connected by the longest stroke in I_1. Rename the nodes associated with the longest stroke in M_2 to L_1 and L_2.
4. Repeat step 3 for the shortest stroke.
5. Identify unused node indices from M_1 and unnamed nodes in M_2. Assign the indices to the nodes.
6. For nodes that have multiple possible indices, create permutations of M_2 for each possibility. For each permutation, compute the dot product as before and store the largest dot product.

Although this method reduced the permutation search space significantly, it presented challenges in comparing complex images. Results with a small subset of categories did not show any improvement over previous results.

5.3 Interpretation

Defining a coordinate-agnostic representation that captured how skeletons were drawn was the first and foundational goal of our analysis. To that effect, we experimented with three candidates - program IDs, grids, adjacency matrices.

The first, program IDs, were not discriminative. A potential cause of this is the simplification of the drawings. We had selected six functions, while in reality, the drawing process was much more complicated. Increasing the amount of information in program IDs could take them in the direction of improved discrimination.
The second, grid-based representations, failed because they were not coordinate-agnostic.
The third, adjacency matrices, did not capture the drawing process but the skeletal structure. Two skeleton matrices could be made to look the same by permutation, as long as they had the same number of nodes and edges. Factoring in the extent of permutation could improve the analysis.
However, in our opinion, the bigger problem lay in the inference of motor programs. The BPL framework, in its current form, knows how to write alphabets. But the primitive structures involved in writing alphabets are significantly different from those involved in drawing skeletons. As a result, BPL's inference does not reflect human drawings accurately. There is also degeneracy in the inference, meaning that the program infers different drawings for the same image. In order to truly capture how humans would draw these images, BPL should be retrained on a more appropriate dataset.
In conclusion, although our representations failed to discriminate between categories in their current form, we believe it is a technical failure rather than a theoretical one, and technical improvements can produce successful discriminatory behavior.

Chapter 6

Conclusion and Discussion

We began this report by discussing artificial object recognition and how it is different from its human counterpart. We aimed to bring the two closer to each other. In an ideal world, we would know how humans recognize objects and replicate the process in synthetic systems. Unfortunately, our knowledge of biological recognition is limited. An intuitive way to look at recognition is through the lens of representations and concepts. We store concepts in our minds/brains, represent incoming information in a format similar to that of concepts, and compare the two. What could this 'standard format' be? We looked at several candidates - imagery, symbols, neural activations - and settled on structural descriptions. The fundamental difference between our formulation of structure and the traditional one is this - traditional approaches describe object structures visually while we described them in terms of motor affordance.

We created motor representations of visual objects wherein we described how humans drew the objects (we called these descriptions 'motor programs'). To compare motor programs and compute the similarity between objects, we defined and tested three metrics. We have already discussed these metrics in detail in the previous chapter. The critical thing to note here is that there are two ways to interpret our hypothesis - the 'hard' interpretation is an embodied take on visual perception. It claims that the brain infers motor programs and compares them. That is not the interpretation we tested in this work. We tested the 'soft' interpretation that a motor affordance-based representation can improve artificial object recognition. Although our experiments did not support this, we did not interpret it as a falsification of the hypothesis itself. If the hypothesis *is* true, it can lead to questions about whether different representations can produce the same behavior and, conversely, if cognition (or at least some part of it) is representation-invariant, after all.

Appendix A

Data

A.1 Category Labels

Category labels were taken from 3 datasets.

A.1.1 CIFAR-100

Baby, Man, Whale, Dolphin, Shark, Trout, Tulip, Sunflower, Bottle, Plate, Pear, Mushroom, Telephone, Television, Chair, Table, Cockroach, Butterfly, Snail, Bear, Wolf, House, Mountain, Camel, Chimpanzee, Kangaroo, Raccoon, Fox, Spider, Crab, Lizard, Turtle, Crocodile, Rabbit, Mouse, Squirrel, Tree, Bus, Bike, Rocket, Lawn-mower

A.1.2 CIFAR-10

Cat, Bird, Kite, Frog

A.1.3 ImageNet

Hornbill, Guitar, Hen, Snake, Umbrella

FIGURE A.1: Dataset preview

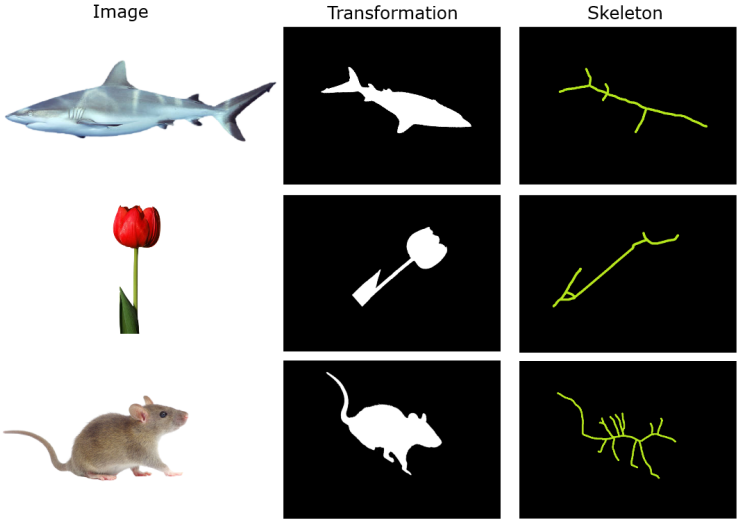


FIGURE A.2: Images (left column) were binarized, inverted and transformaed (middle) and then skeletonized (right).

A.2 Transformation and Skeletonization

We removed all background information from the images such that only the object was visible in the picture. Then, we inverted and binarized all images.
For rotational transforms, we chose a rotation parameter $\theta \in [0, 360]$ randomly. For scalar transforms, we chose a scaling parameter $s \in [0.5, 2]$ randomly. After transformation, we skeletonized the images using Zhang's Method[58] (Figure A.2).

Bibliography

[1] Alcorn, M., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.-S., and Nguyen, A. (2019). Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. pages 4840–4849.

[2] Aloimonos, J. and Shulman, D. (1989). *Integration of visual modules: An extension of the Marr paradigm*. Academic Press.

[3] Ayzenberg, V., Chen, Y., and Yousif, S. (2019). Skeletal representations of shape in human vision: Evidence for a pruned medial axis model. *Journal of Vision*, 19.

[4] Ayzenberg, V. and Lourenco, S. F. (2019). Skeletal descriptions of shape provide unique perceptual information for object recognition. *Scientific Reports*, 9:9359.

[5] Baker, N., Lu, H., Erlikhman, G., and Kellman, P. (2018). Deep convolutional networks do not classify based on global object shape. *PLOS Computational Biology*, 14:e1006613.

[6] Baker, N., Lu, H., Erlikhman, G., and Kellman, P. (2020). Local features and global shape information in object classification by deep convolutional neural networks. *Vision Research*, 172:46–61.

[7] Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147.

[8] Blake, A. and Zisserman, A. (1987). Visual reconstruction.

[9] Brincat, S. L. and Connor, C. E. (2004). Underlying principles of visual shape selectivity in posterior inferotemporal cortex. *Nature Neuroscience*, 7:880–886.

[10] Carlson, T., Tovar, D. A., Alink, A., and Kriegeskorte, N. (2013). Representational dynamics of object vision: The first 1000 ms. *Journal of Vision*, 13(10):1–1.

[11] Choi, H. I. and Han, C. Y. (2002). Chapter 19 - the medial axis transform. In Farin, G., Hoschek, J., and Kim, M.-S., editors, *Handbook of Computer Aided Geometric Design*, pages 451–471. North-Holland, Amsterdam.

[12] Cummins, R. (1991). *Meaning and Mental Representation*. Bradford Books.

[13] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database.

[14] Desimone, R., Albright, T., Gross, C., and Bruce, C. (1984). Stimulus-selective properties of inferior temporal neurons in the macaque. *J. Physiol*, 357:219–240.

[15] DiCarlo, J. J. and Cox, D. D. (2007). Untangling invariant object recognition. *Trends in Cognitive Sciences*, 11(8):333–341.

[16] DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415–434.

[17] Edelman, S. (1999). *Representation and Recognition in Vision*. Bradford Books.

[18] Edelman, S. and Weinshall, D. (1994). Computational approaches to shape constancy.

[19] Elder, J. H. (2018). Shape from contour: Computation and representation. *Annual Review of Vision Science*, 4(1):423–450. PMID: 30222530.

[20] Elder, J. H. and Velisavljevic, L. (2009). Cue dynamics underlying rapid detection of animals in natural scenes. *Journal of Vision*, 9(8).

[21] Ellis, K. (2020). *Algorithms for Learning to Induce Programs*. PhD thesis, Massachusetts Institute of Technology.

[22] Erdogan, G. and Jacobs, R. (2017). Visual shape perception as bayesian inference of 3d object-centered shape representations. *Psychological review*, 124.

[23] Feldman, J. and Singh, M. (2006). Bayesian estimation of the shape skeleton. *Proceedings of the National Academy of Sciences of the United States of America*, 103:18014–9.

[24] Gelder, T. V. (1995). What might cognition be if not computation? *Journal of Philosophy*, 92(7):345–81.

[25] Giblin, P. and Kimia, B. (2003). On the intrinsic reconstruction of shape from its symmetries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:895–911.

[26] Gross, C. G., Rocha-Miranda, C. E., and Bender, D. B. (1972). Visual properties of neurons in inferotemporal cortex of the macaque. *Journal of Neurophysiology*, 35(1):96–111.

[27] Hebart, M. N., Zheng, C. Y., Pereira, F., and Baker, C. I. (2020). Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. *Nature Human Behaviour*, 4:1173–1185.

[28] Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243.

[29] Hung, C.-c., Carlson, E., and Connor, C. (2012). Medial axis shape coding in macaque inferotemporal cortex. *Neuron*, 74:1099–113.

[30] Jacob, G., Rt, P., Katti, H., and Arun, S. (2021). Qualitative similarities and differences in visual object representations between brains and deep networks. *Nature Communications*, 12.

[31] Kayaert, G., Biederman, I., Op de Beeck, H. P., and Vogels, R. (2005). Tuning for shape dimensions in macaque inferior temporal cortex. *European Journal of Neuroscience*, 22(11):212–224.

[32] Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10:e1003915.

[33] Kovacs, I., Feher, A., and Julesz, B. (1998). Medial-point description of shape: A representation for action coding and its psychophysical correlates. *Vision research*, 38:2323–33.

[34] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

[35] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25.

[36] Kubilius, J., Bracci, S., and Op de Beeck, H. (2016). Deep neural networks as a computational model for human shape sensitivity. *PLoS computational biology*, 12:e1004896.

[37] Lake, B., Salakhutdinov, R., and Tenenbaum, J. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350:1332–1338.

[38] Lee, T. (2003). Computations in the early visual cortex. *Journal of physiology, Paris*, 97:121–39.

[39] Liu, T.-L. and Geiger, D. (1999). Approximate tree matching and shape similarity. volume 1, pages 456 – 462.

[40] Logothetis, N., Pauls, J., Bülthoff, H., and Poggio, T. (1994). View-dependent object recognition by monkeys. *Current Biology*, 4(5):401–414.

[41] Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395.

[42] Mishkin, M. and Ungerleider, L. G. (1982). Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural Brain Research*, 6(1):57–77.

[43] Navarro, G. (2000). A guided tour to approximate string matching. *ACM Computing Surveys*, 33.

[44] Perrett, D., Hietanen, J., Oram, M., and Benson, P. (1992). Organization and functions of cells responsive to faces in the temporal cortex [and discussion]. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 335:23–30.

[45] Pylyshyn, Z. (1973). What the mind's eye tells the mind's brain: A critique of mental imagery. *Psychological Bulletin*, 80:1–24.

[46] Quian, R., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, 435:1102–7.

[47] Rupert, R. (2013). *The Sufficiency of Objective Representation.*

[48] Sebastian, T., Klein, P., and Kimia, B. (2004). Recognition of shapes by editing shock graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26:550 – 571.

[49] Seibert, D., Yamins, D., Ardila, D., Hong, H., DiCarlo, J. J., and Gardner, J. L. (2016). A performance-optimized model of neural responses across the ventral visual stream. *bioRxiv*.

[50] Suppes, P., Pavel, M., and Falmagne, J. C. (1994). Representations and models in psychology. *Annual Review of Psychology*, 45(1):517–544.

[51] Tarr, M. and Bülthoff, H. (1996). Is human object recognition better described by geon structural descriptions or by multiple views? *Journal of experimental psychology. Human perception and performance*, 21:1494–505.

[52] Tarr, M. and Bülthoff, H. (1998). Image-based object recognition in man, monkey and machine. *Object recognition in man, monkey, and machine*, pages 1–20.

[53] Tarr, M., Williams, P., Hayward, W., and Gauthier, I. (1998). Three-dimensional object recognition is viewpoint dependent. *Nature neuroscience*, 1:275–7.

[54] Trinh, N. and Kimia, B. (2011). Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *International Journal of Computer Vision*, 94:215–240.

[55] Ullman, S. (1989). Aligning pictorial descriptions: an approach to object recognition. *Cognition*, 32(3):193–254.

[56] Wilder, J., Feldman, J., and Singh, M. (2011). Superordinate shape classification using natural shape statistics. *Cognition*, 119:325–40.

[57] Yamins, D., Hong, H., Cadieu, C., Solomon, E., Seibert, D., and Dicarlo, J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 111.

[58] Zhang, T. Y. and Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239.

[59] Zoccolan, D., Kouh, M., Poggio, T., and Dicarlo, J. (2007). Trade-off between object selectivity and tolerance in monkey inferotemporal cortex. *The Journal of Neuroscience*, 27:12292–307.