

Categories and Concepts

Computational models (part 2)

Brenden Lake

PSYCH-GA 2207

The Adaptive Nature of Human Categorization

John R. Anderson
Carnegie Mellon University

A rational model of human categorization behavior is presented that assumes that categorization reflects the derivation of optimal estimates of the probability of unseen features of objects. A Bayesian analysis is performed of what optimal estimations would be if categories formed a disjoint partitioning of the object space and if features were independently displayed within a category. This Bayesian analysis is placed within an incremental categorization algorithm. The resulting rational model accounts for effects of central tendency of categories, effects of specific instances, learning of linearly nonseparable categories, effects of category labels, extraction of basic level categories, base-rate effects, probability matching in categorization, and trial-by-trial learning functions. Although the rational model considers just 1 level of categorization, it is shown how predictions can be enhanced by considering higher and lower levels. Considering prediction at the lower, individual level allows integration of this rational analysis of categorization with the earlier rational analysis of memory (Anderson & Milson, 1989).

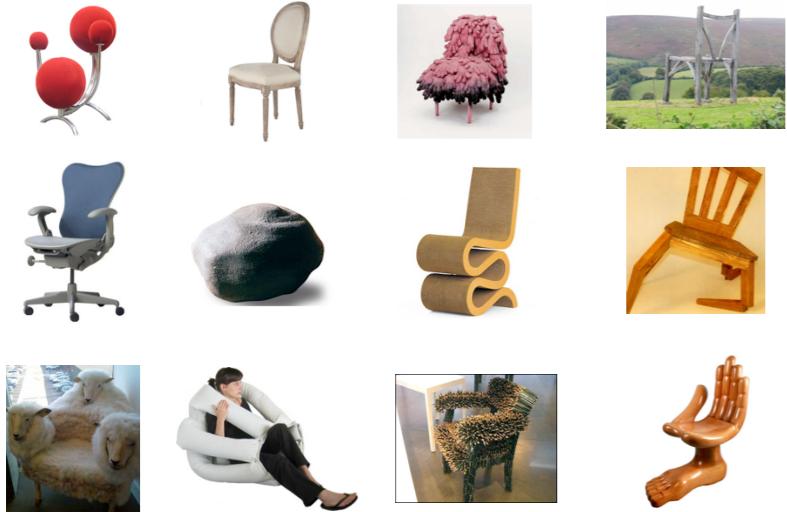
Anderson (1990) presented a rational analysis of human cognition. The term *rational* derives from similar "rational-man" analyses in economics. Rational analyses in other fields are sometimes called *adaptationist analyses*. Basically, they are efforts to explain the behavior in some domain on the assumption that the behavior is optimized with respect to some criteria of adaptive importance. This article begins with a general characterization of how one develops a rational theory of a particular cognitive phenomenon. Then I present the basic theory of categorization developed in Anderson (1990) and review the applications from that book. Since the writing of the book, the

steps involved in a research program that attempts to understand cognition in terms of its adaptation to the environment:

1. The first task is to specify what the system is trying to optimize. Perhaps such models are ultimately to be justified in terms of maximizing some evolutionary criterion like number of surviving offspring. However, this is not a very workable criterion in most applications. Thus, economics uses wealth as the variable to be optimized; optimal foraging theory (Stephens & Krebs, 1986) often uses caloric intake; and the rational theory of memory (Anderson & Milson, 1989) uses retrieval of relevant experiences from the past.

Is there another option besides learning a single prototype, or storing every single exemplar?

Exemplars



Another option?



Prototype



Somewhere between exemplars and prototypes...

The mind could represent a concept, like “boat”, as a collection of *several* prototypes... and even specific examples.

We want **model flexibility** depending on the **structure of the data**.

prototype 1



prototype 2

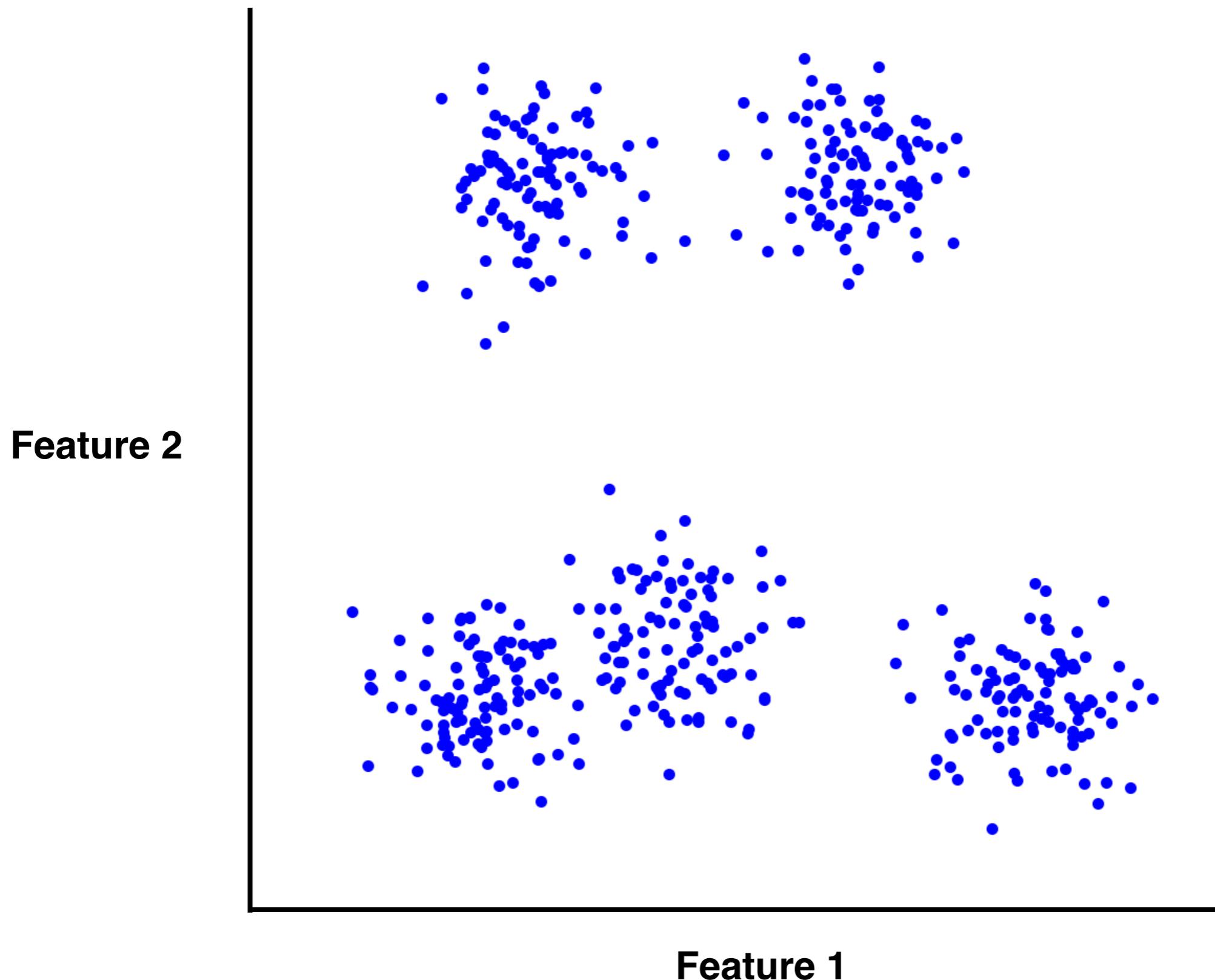


prototype 3

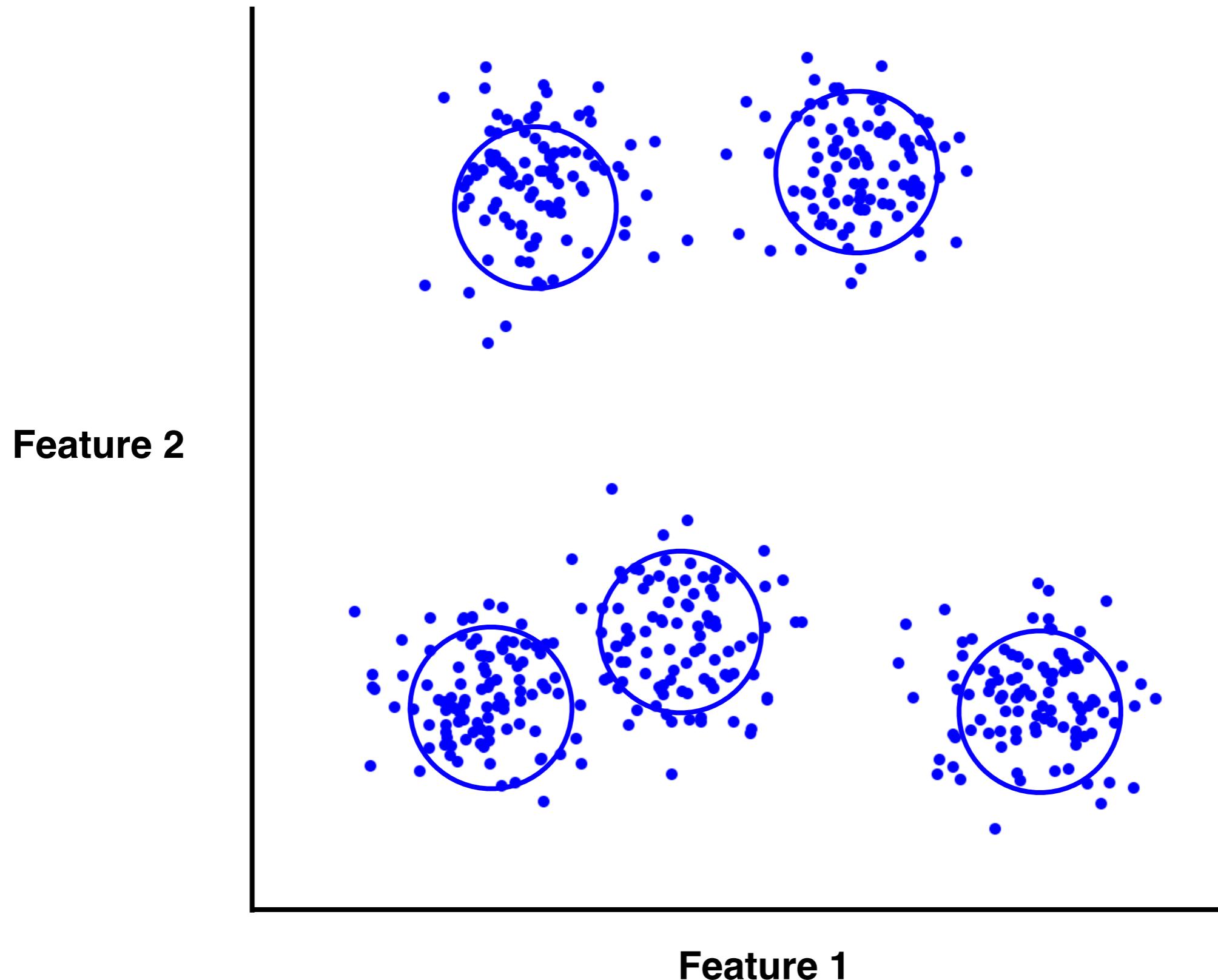


...

Stimuli plotted in feature space (possibly many dimensions)



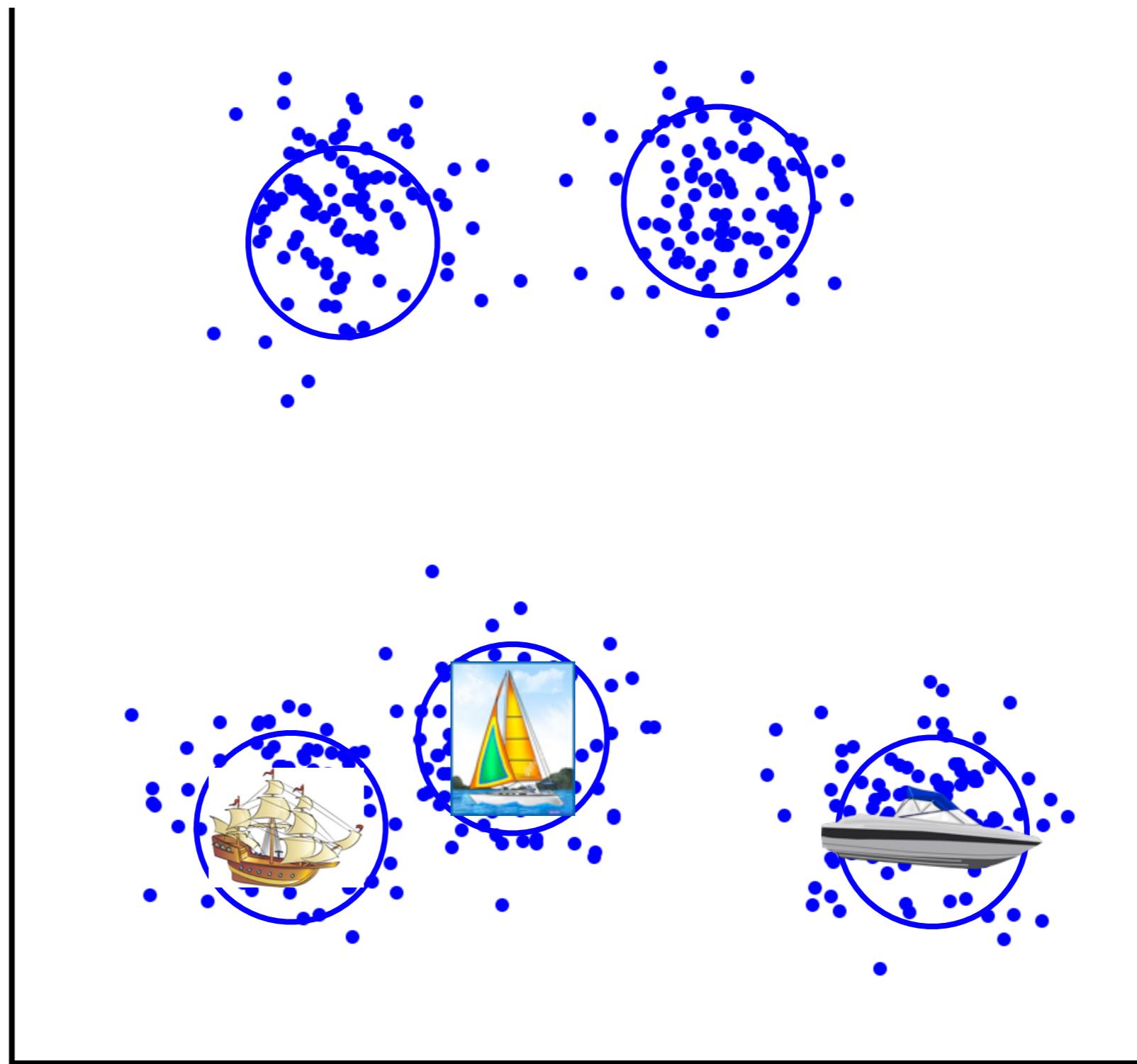
Stimuli plotted in feature space



Stimuli plotted in feature space

Feature 2

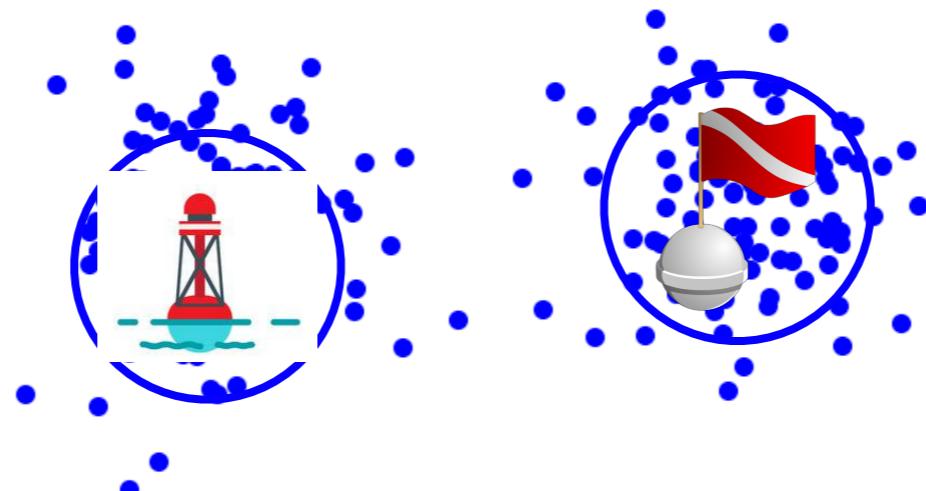
Feature 1



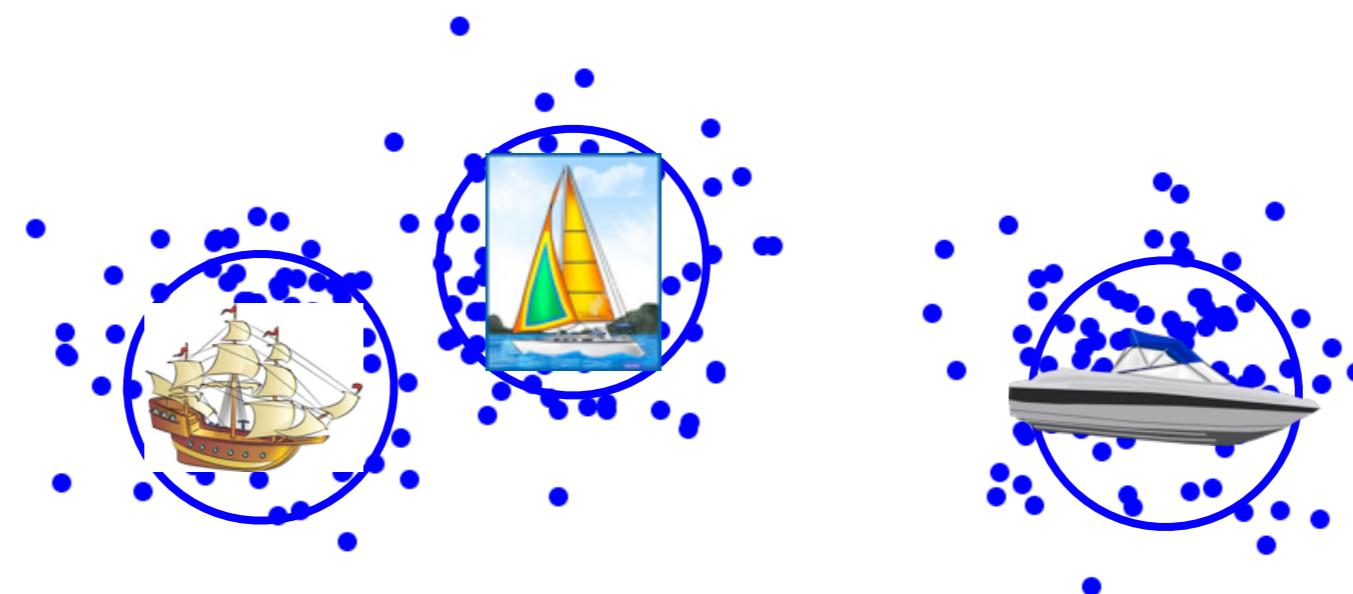
Stimuli plotted in feature space

Feature 2
("class label")

Feature 1

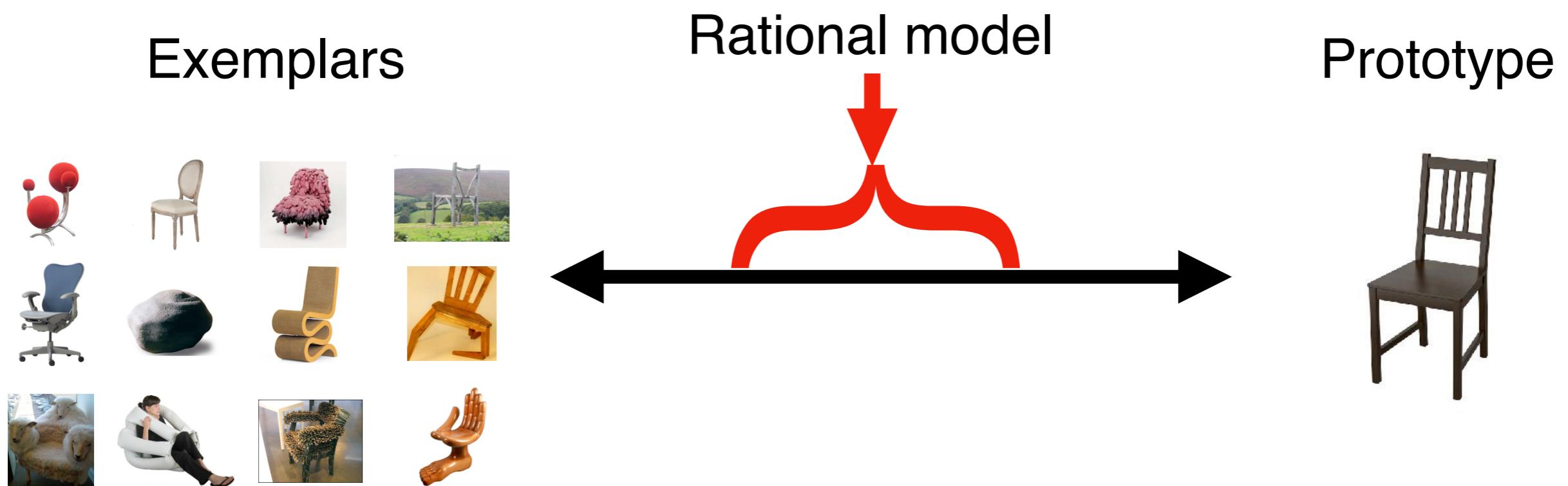


2 categories
("boat" vs. "buoy")



5 clusters

The Rational model can discover cluster structure, interpolating between prototype and exemplar models in a principled way



Anderson's “rational model”

Base rate of CLUSTER k (not category) is the probability of a new stimulus belonging to that cluster

(this model is called the “Chinese Restaurant Process” or “Dirichlet Process” in statistics and machine learning)

Base rate assignment to existing cluster k

$$P(k) = \frac{n_k}{\alpha + n}$$

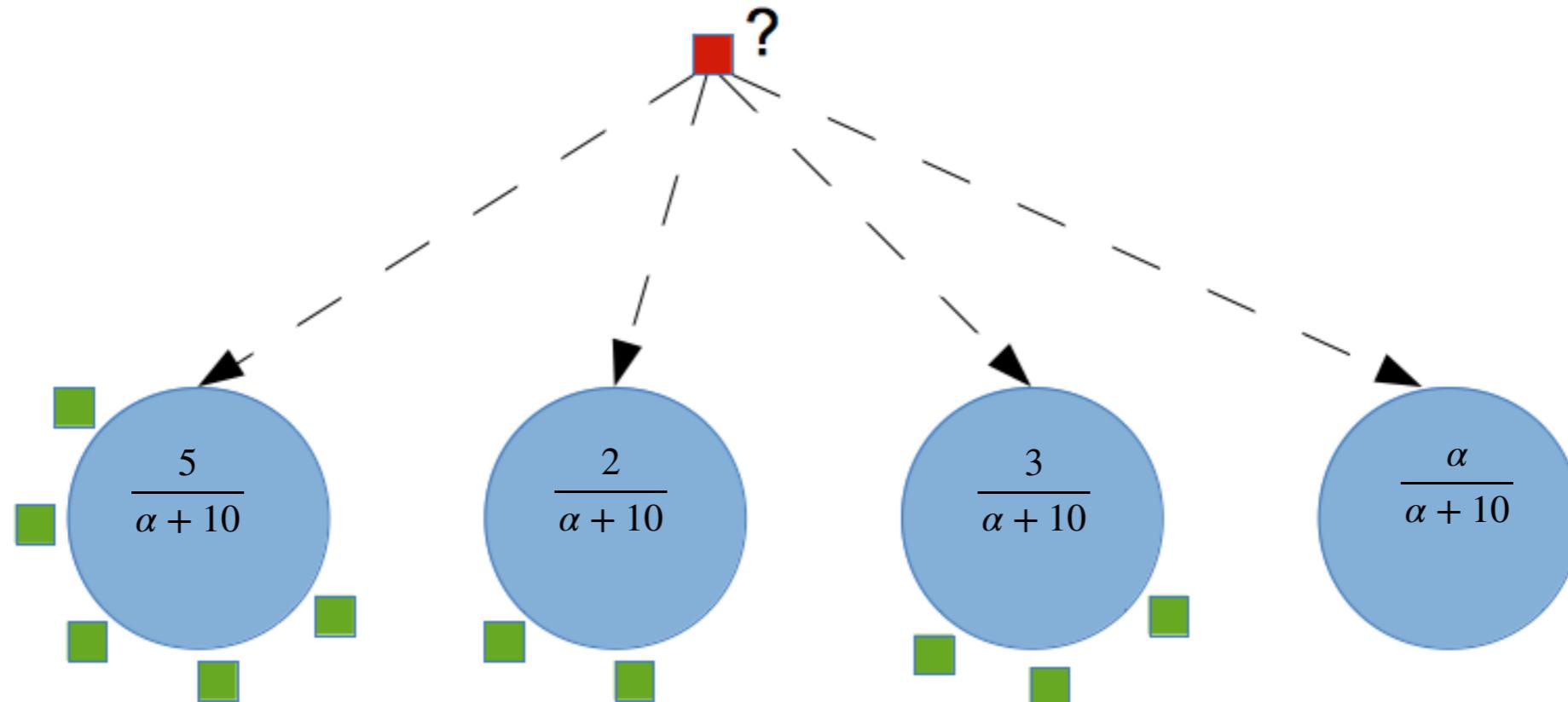
Base rate of forming a new cluster

$$P(\text{new}) = \frac{\alpha}{\alpha + n}$$

n_k number of stimuli in k

n number of stimuli total

α parameter that controls likelihood of new cluster



Rational model

Base rate of CLUSTER k (not category) probability for new stimuli (prior)

(this model is called the “Chinese Restaurant Process” or “Dirichlet Process” in statistics and machine learning)

Base rate assignment to existing cluster k

$$P(k) = \frac{n_k}{\alpha + n}$$

Base rate of forming a new cluster

$$P(\text{new}) = \frac{\alpha}{\alpha + n}$$

n_k number of stimuli in k

n number of stimuli total

α parameter that controls likelihood of new cluster

Scoring similarity of new stimulus F to existing cluster k (likelihood)

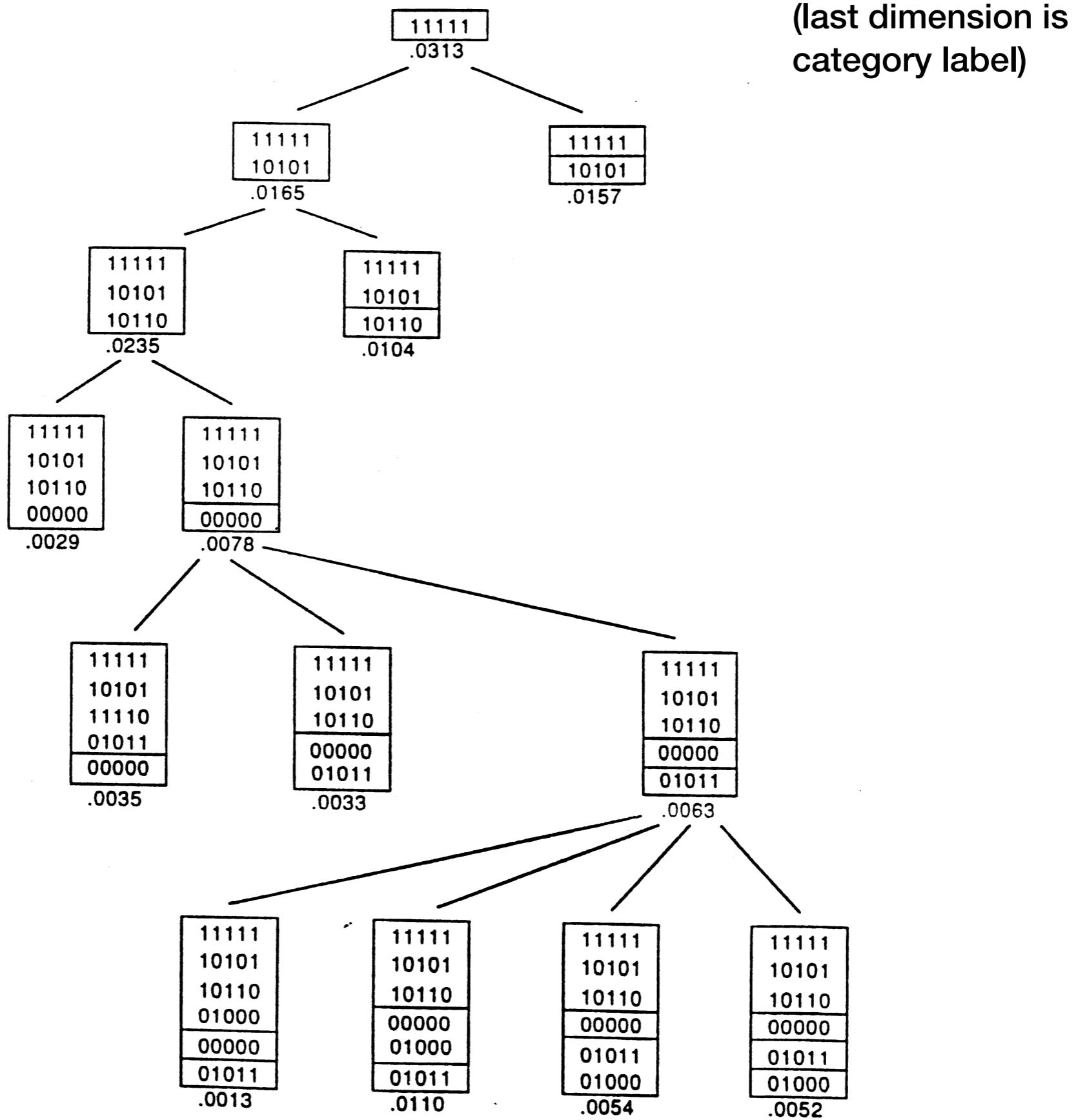
$$P(F|k) = \prod_{D_i} P(F_i|k)$$

The form of this distribution depends on whether features are discrete or continuous. **Class labels** are treated just like any other feature (see paper for details)

Using Bayes’ rule, we to classify the new item F in a cluster k (posterior)

$$P(k|F) = \frac{P(F|k)P(k)}{P(F|\text{new})P(\text{new}) + \sum_{k'} P(F|k')P(k')}$$

Rational model in action



Rational model in action

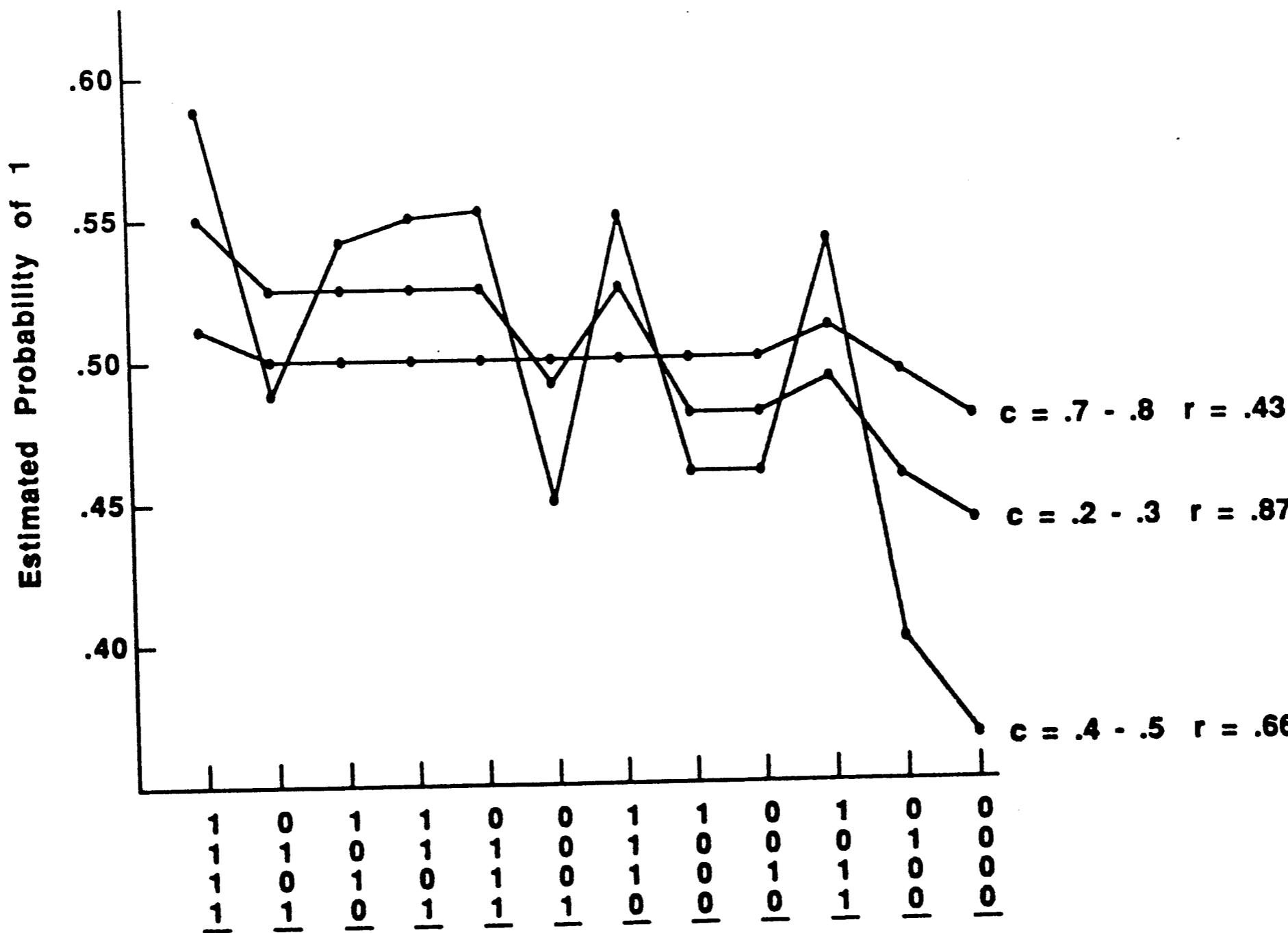
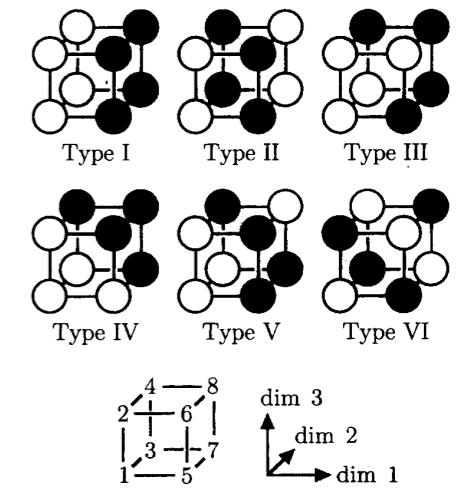


Figure 2. Estimated probability of Category 1 for the 12 test stimuli in the first experiment of Medin and Schaffer (1978). (Different functions are for different ranges of the coupling probability.)

Comparing SHJ learning curves for people and rational model



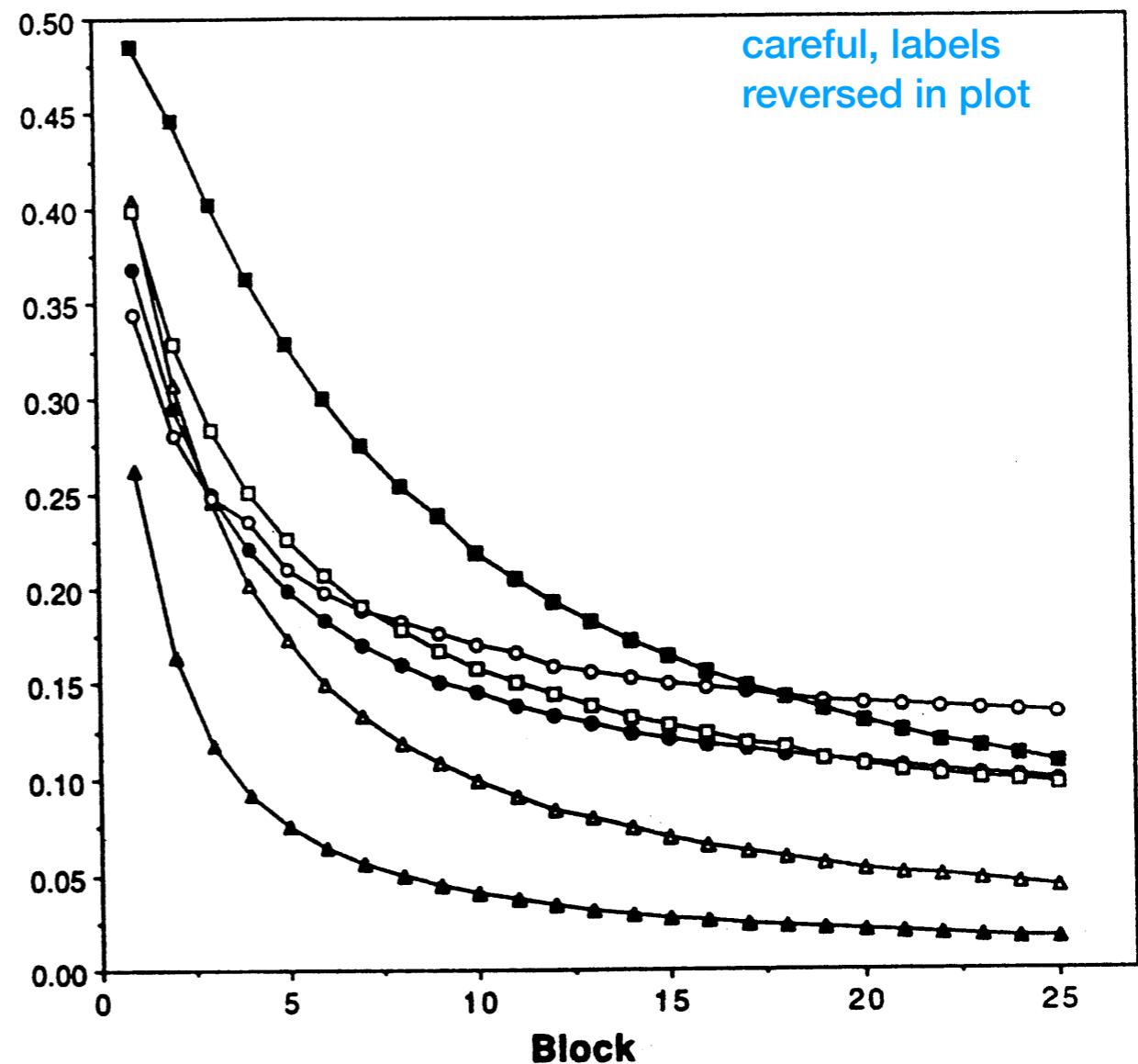
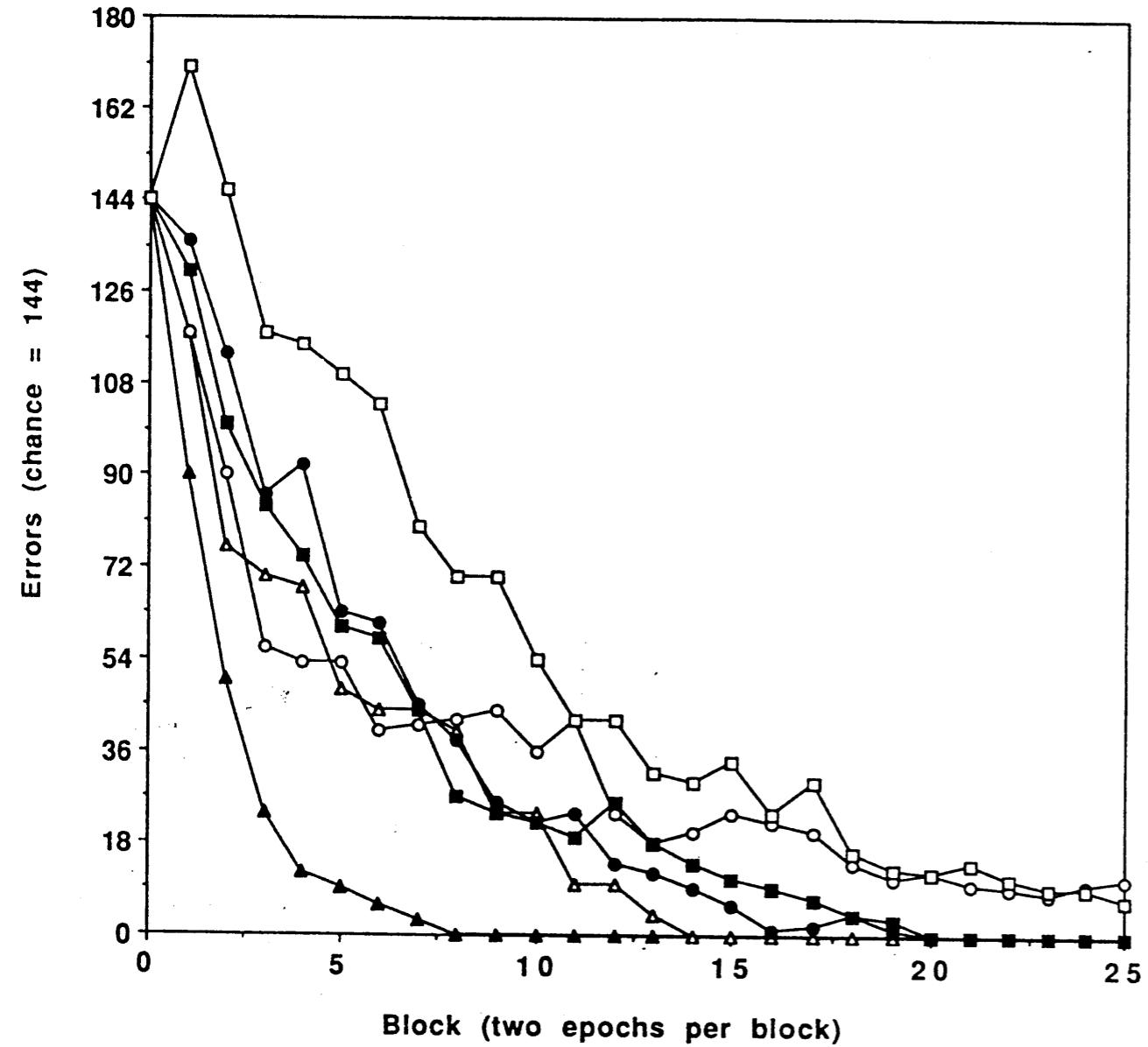
People

▲ Type I ● Type III ■ Type V
▲ Type II ○ Type IV □ Type VI

Model

▲ Type I ● Type III
▲ Type II ○ Type IV
□ Type V ■ Type VI

careful, labels reversed in plot



ImageNet Classification with Deep Convolutional Neural Networks

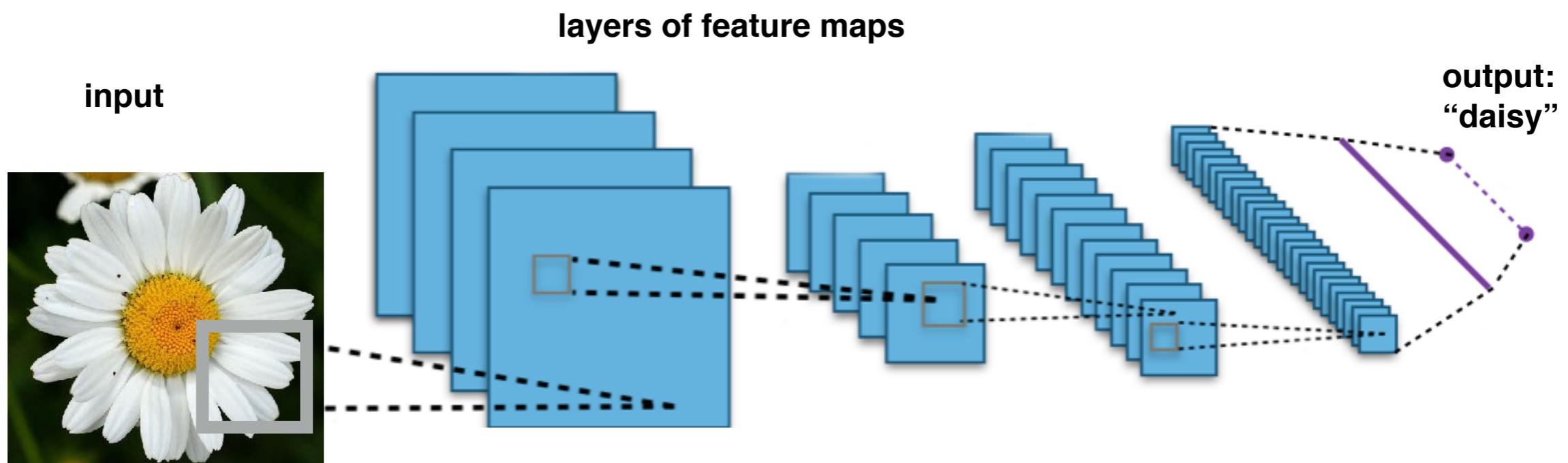
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

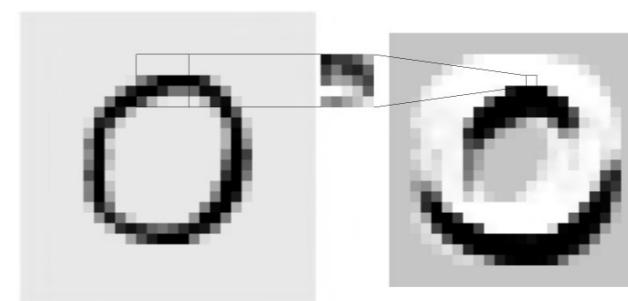
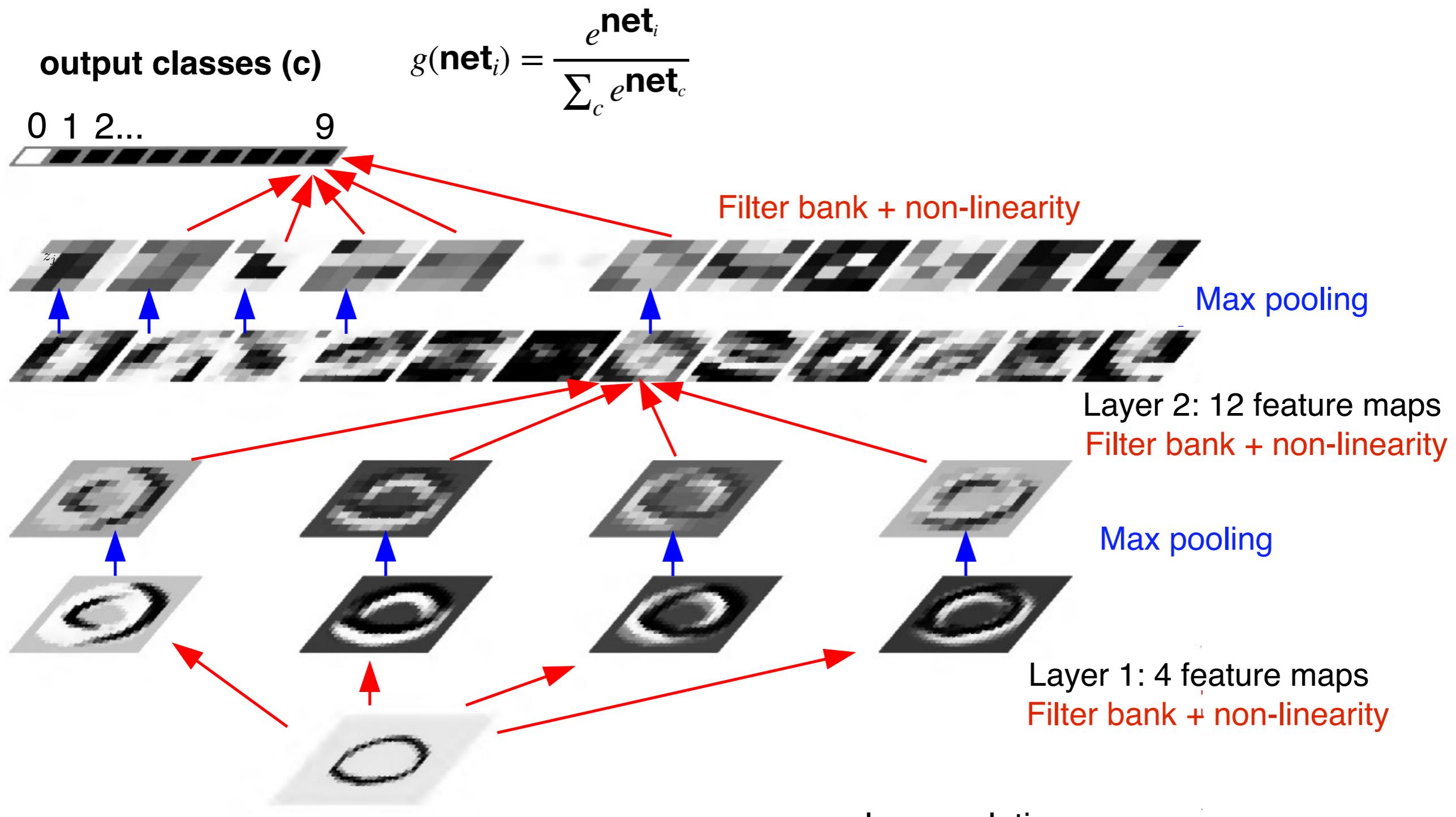


convolving an image with a filter

We slide the filter across the image to produce an output image

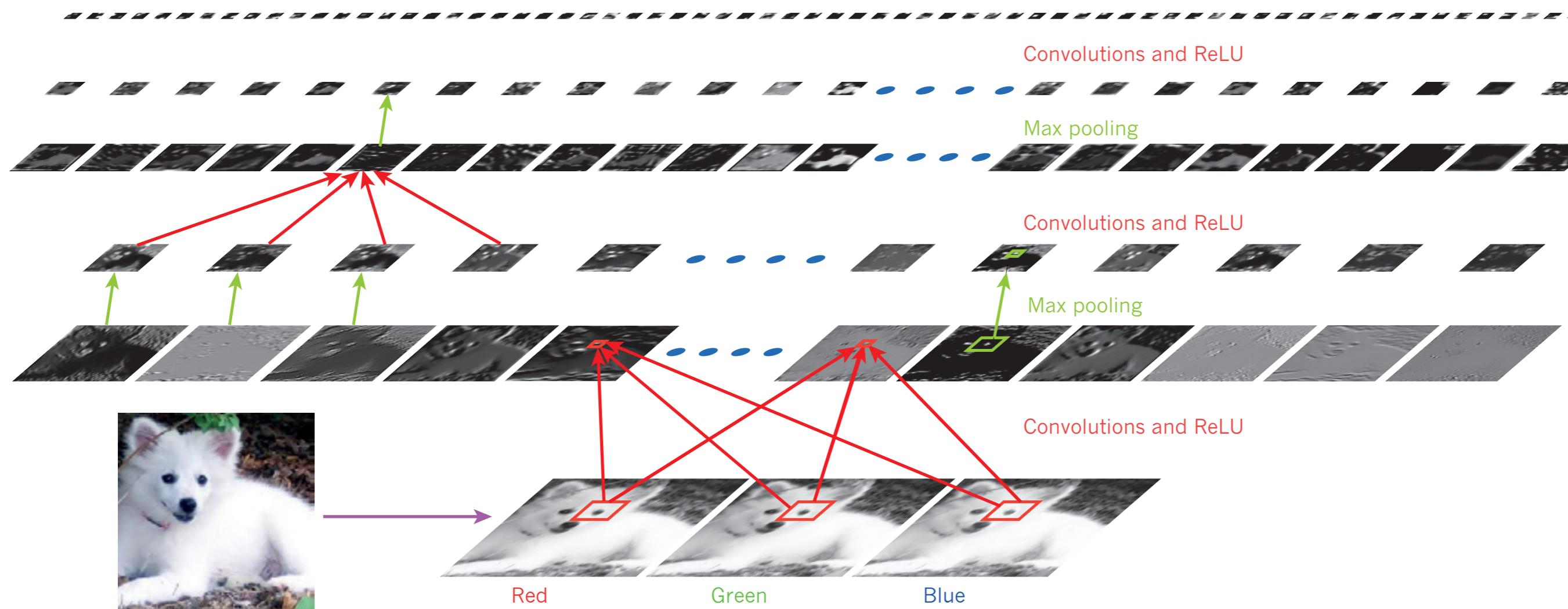
image	filter	output																																																		
<table border="1" style="border-collapse: collapse; width: 100%;"><tr><td>22</td><td>15</td><td>1</td><td>3</td><td>60</td></tr><tr><td>42</td><td>5</td><td>38</td><td>39</td><td>7</td></tr><tr><td>28</td><td>9</td><td>4</td><td>66</td><td>79</td></tr><tr><td>0</td><td>82</td><td>45</td><td>12</td><td>17</td></tr><tr><td>99</td><td>14</td><td>72</td><td>51</td><td>3</td></tr></table>	22	15	1	3	60	42	5	38	39	7	28	9	4	66	79	0	82	45	12	17	99	14	72	51	3	\times	<table border="1" style="border-collapse: collapse; width: 100%;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	15	1	3	60																																																
42	5	38	39	7																																																
28	9	4	66	79																																																
0	82	45	12	17																																																
99	14	72	51	3																																																
0	0	0	0	0																																																
0	0	0	1	0																																																
0	0	0	0	0																																																
0	0	0	0	0																																																
0	0	0	0	0																																																
		$=$																																																		
		<table border="1" style="border-collapse: collapse; width: 100%;"><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>1</td><td>3</td><td>60</td><td></td></tr><tr><td></td><td>38</td><td>39</td><td>7</td><td></td></tr><tr><td></td><td>4</td><td>66</td><td>79</td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>							1	3	60			38	39	7			4	66	79																															
	1	3	60																																																	
	38	39	7																																																	
	4	66	79																																																	

Deep convolutional neural network (convnet) for vision



Deep convolutional neural network

Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)



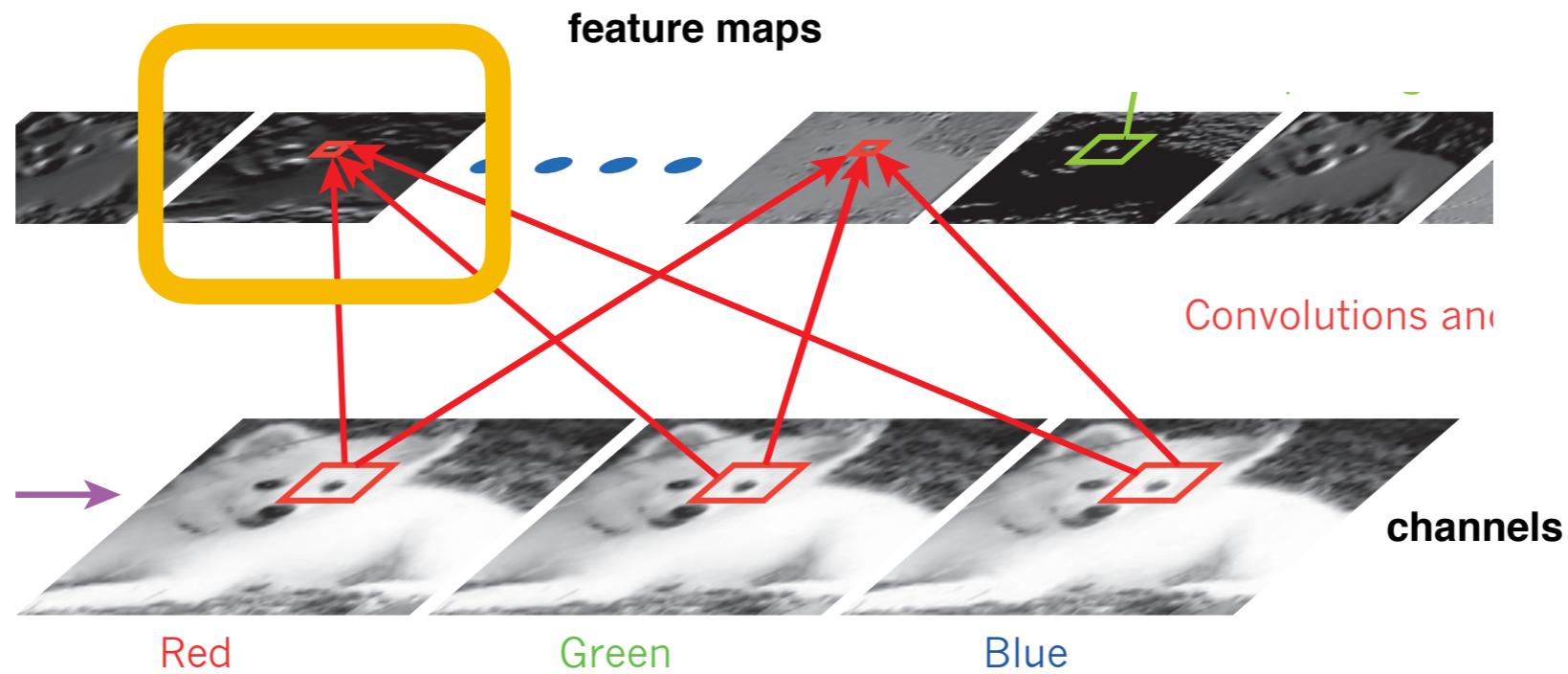
From LeCun, Bengio, & Hinton (2015).

Examples of learned filters



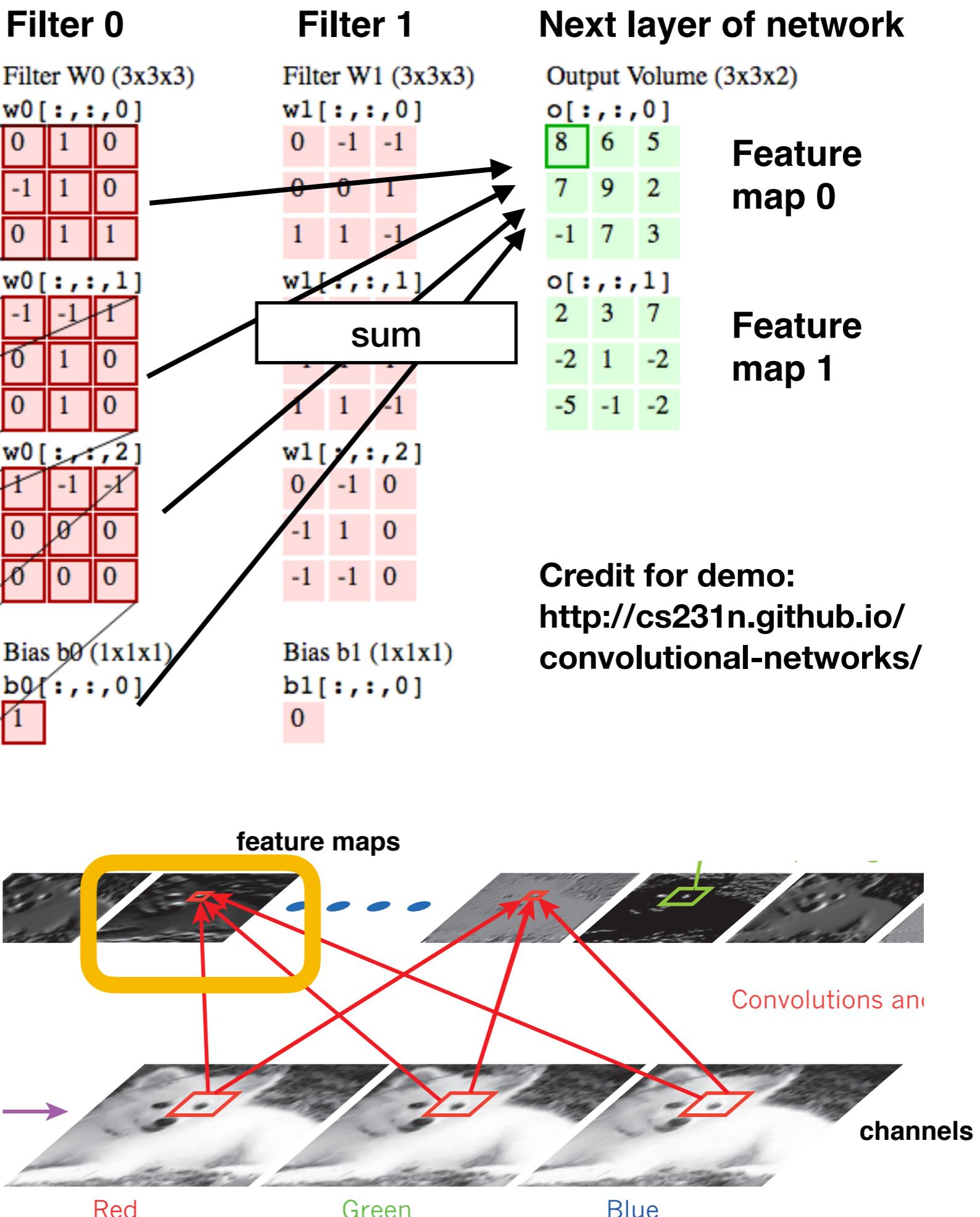
(Krizhevsky et al., 2012)

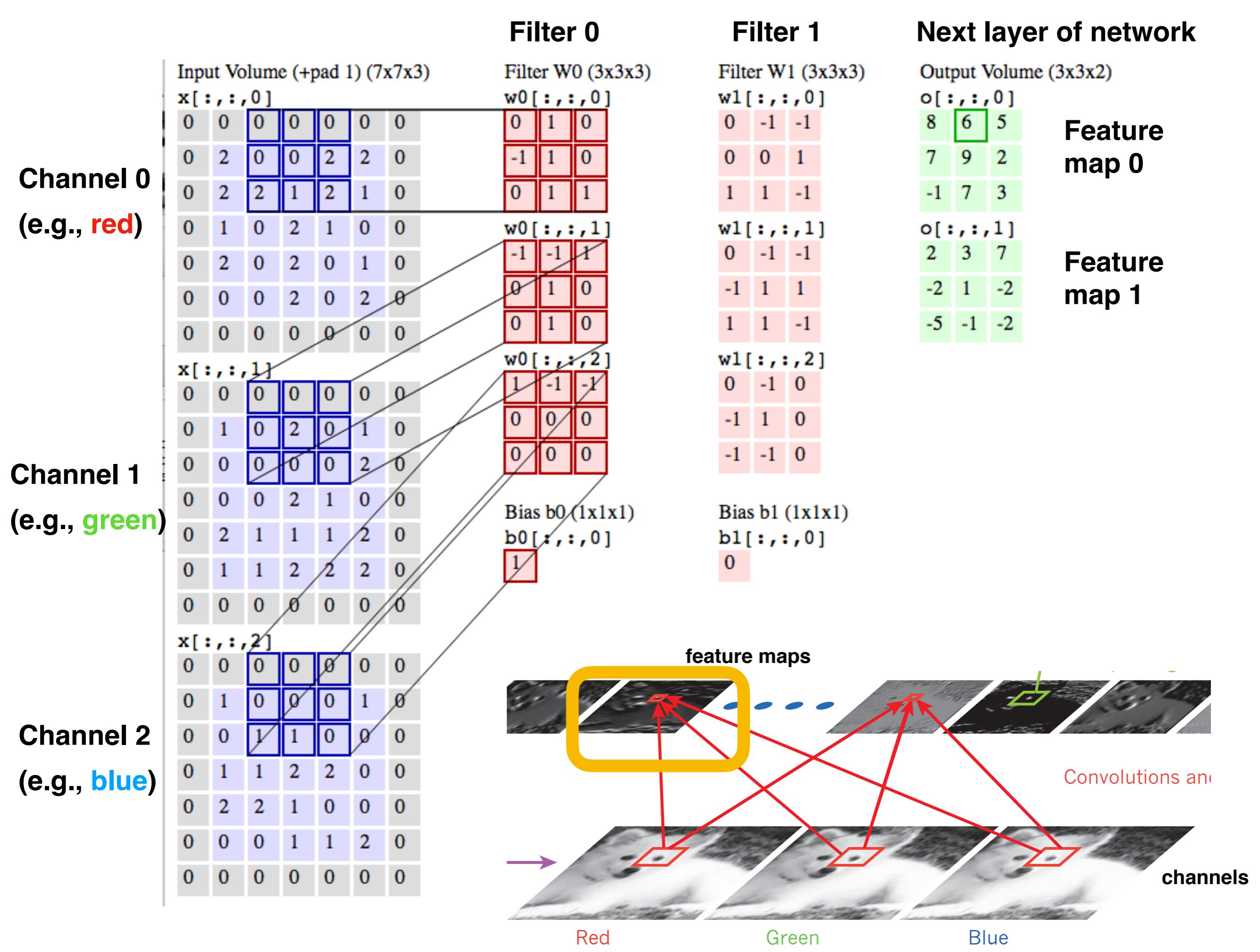
Let's go through computing with convolutions to build a feature map...

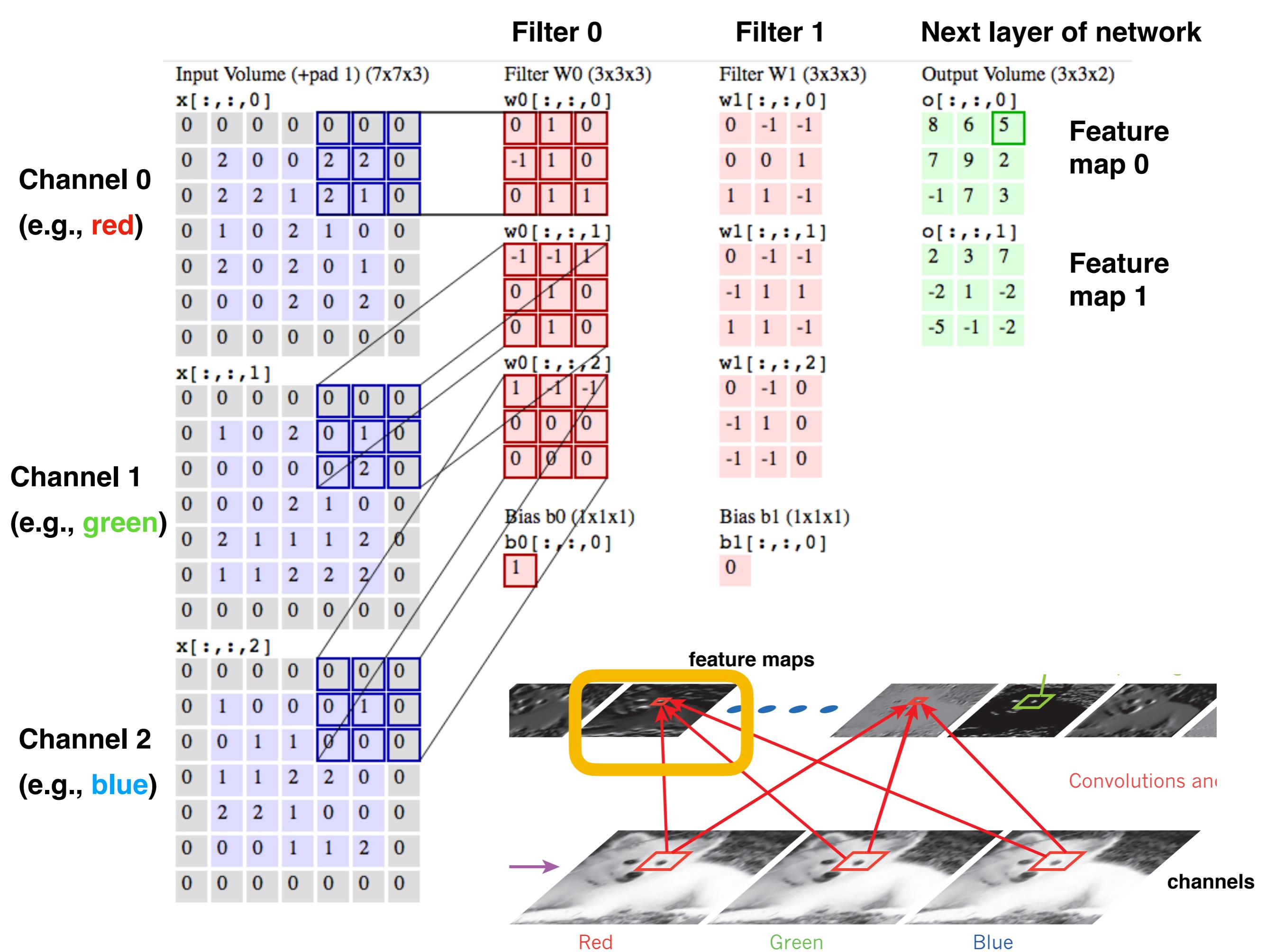


Channel 0
(e.g., red)

	Input Volume (+pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
x[:, :, 0]	$x[:, :, 0]$	$w0[:, :, 0]$	$w1[:, :, 0]$	$o[:, :, 0]$
0 0 0 0 2 0 0 2 2	0 0 0 0 2 0 0 2 2	0 1 0 -1 1 0 0 1 1	0 -1 -1 0 0 1 1 1 -1	8 6 5 7 9 2 -1 7 3
0 1 0 0 2 0 0 0 0	0 1 0 0 2 0 0 0 0	-1 -1 1 0 1 0 0 1 0	1 1 -1 -1 1 0 -1 -1 0	2 3 7 -2 1 -2 -5 -1 -2
x[:, :, 1]	0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0	1 -1 -1 0 0 0 0 0 0	
0 0 0 0 1 0 0 0 0	0 0 0 2 0 1 0 0 0 2 0	0 0 0 0 0 0 0 0 0	0 -1 0 -1 1 0 -1 -1 0	
0 0 0 0 2 1 0 1 1	0 0 0 2 1 0 1 1 2 0	1 1 1 0 0 0 0 0 0		
x[:, :, 2]	0 1 1 0 0 0	1 0	Bias b0 (1x1x1) b0[:, :, 0]	Bias b1 (1x1x1) b1[:, :, 0]
0 0 0 0 0 0	0 0 0 0 0 0 0 0 0		1 0	



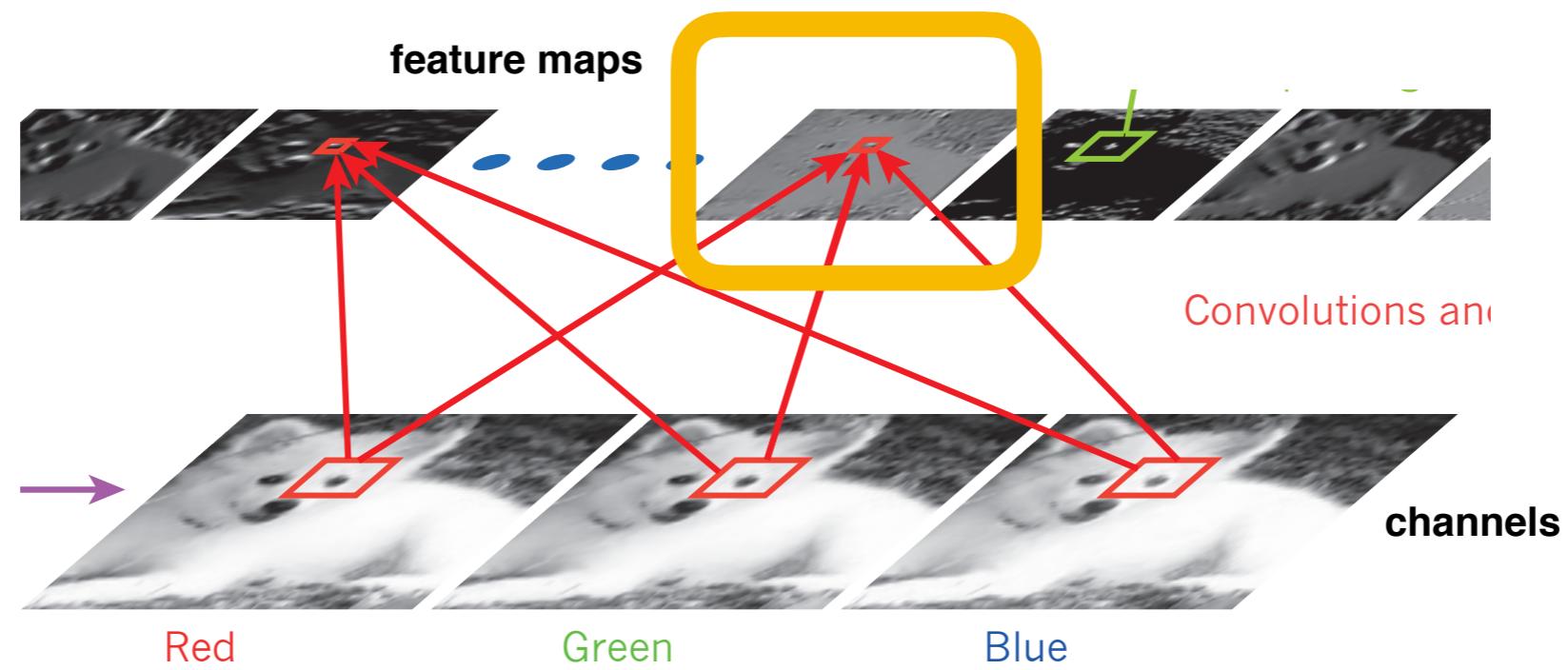




Channel 0
(e.g., red)

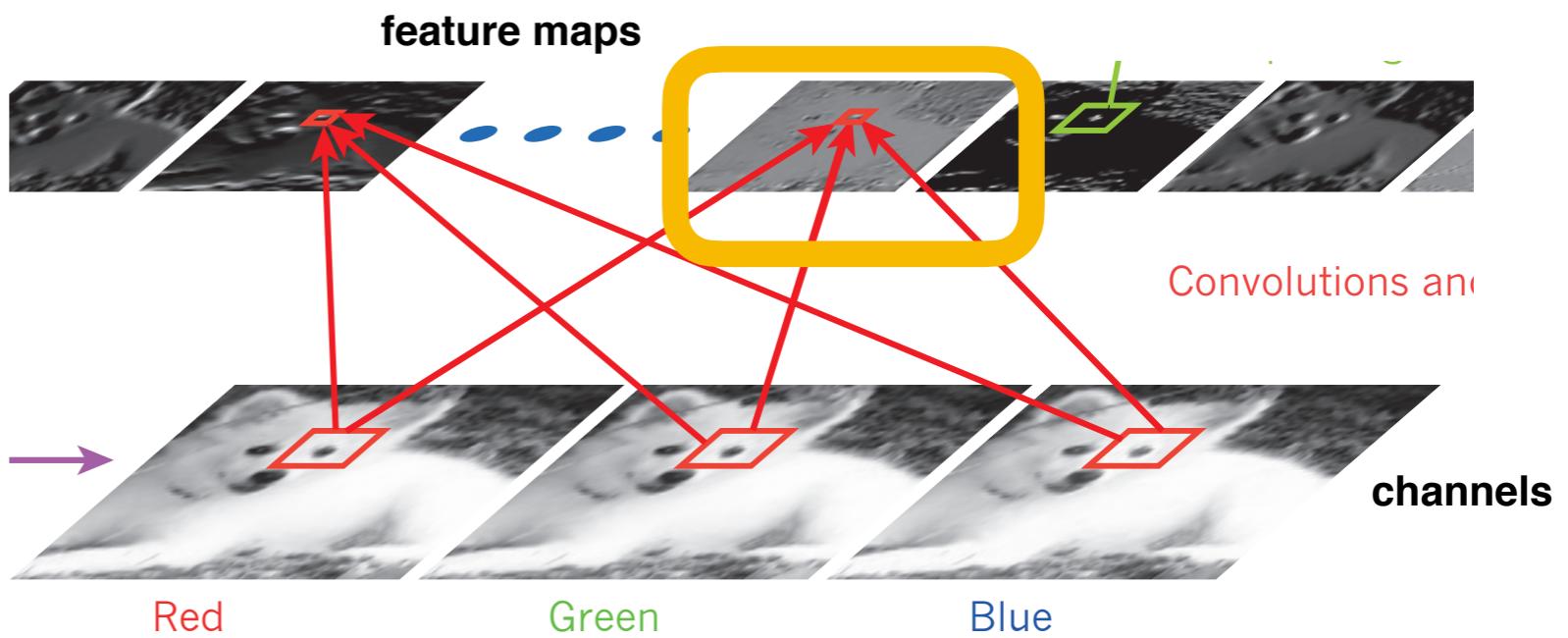
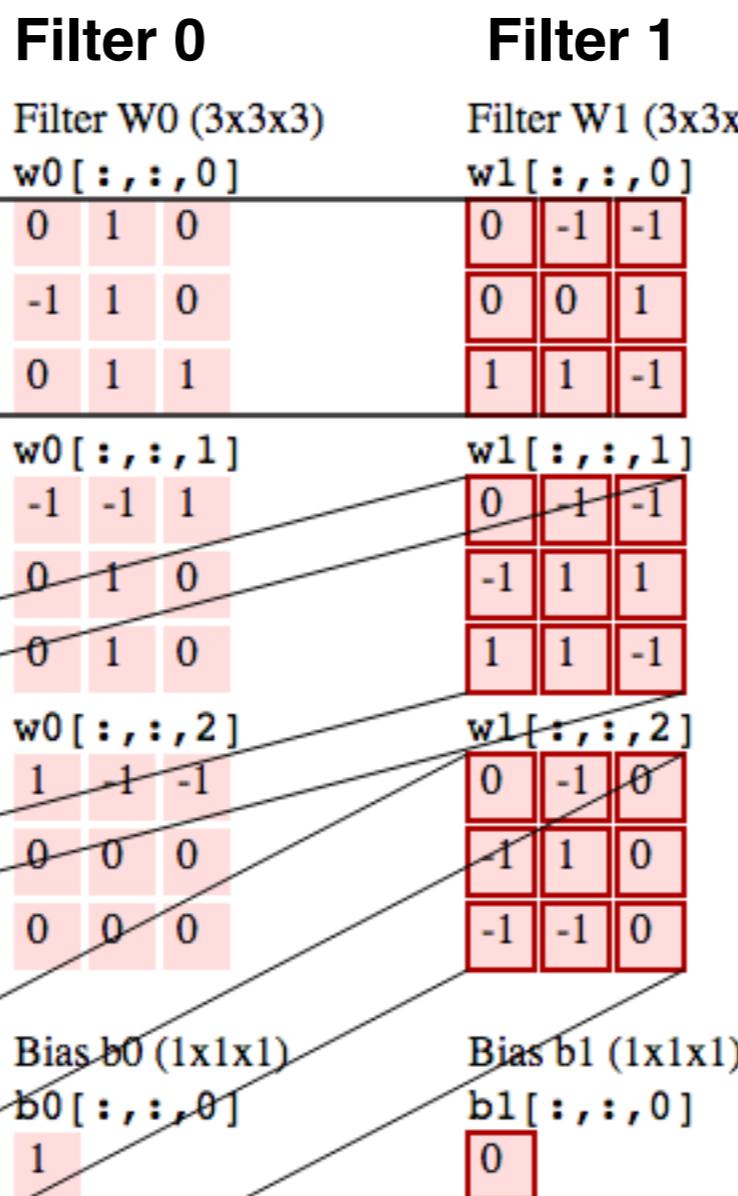
Input Volume (+pad 1) (7x7x3)		Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[:, :, 0]$		$w0[:, :, 0]$	$w1[:, :, 0]$	$o[:, :, 0]$
0 0 0 0 0 0 0		0 1 0 -1 1 0 0 1 1	0 -1 -1 0 0 1 1 1 -1	8 6 5 7 9 2 -1 7 3
0 2 0 0 2 2 0		-1 1 0 0 1 1 0 1 0	w1[:, :, 1]	$o[:, :, 1]$
0 2 2 1 2 1 0		-1 -1 1 0 1 0 0 1 0	0 -1 -1 -1 1 1 1 1 -1	2 3 7 -2 1 -2 -5 -1 -2
0 1 0 2 1 0 0		1 -1 -1 0 0 0 0 0 0	w1[:, :, 2]	
0 2 0 2 0 1 0		0 0 0 0 0 0 0 0 0	0 -1 0 -1 1 0 -1 -1 0	
0 0 0 2 0 2 0				
0 0 0 0 0 0 0				
$x[:, :, 1]$				
0 0 0 0 0 0 0				
0 1 0 2 0 1 0				
0 0 0 0 0 2 0				
0 0 0 0 0 0 0				
$x[:, :, 2]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 2 0				
0 1 1 2 2 2 0				
0 0 0 0 0 0 0				
$x[:, :, 3]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 4]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 5]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 6]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 7]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 8]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 9]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 10]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 11]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 12]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 13]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 14]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 15]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 16]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 17]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 18]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 19]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 20]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 21]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 22]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 23]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 24]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 25]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 26]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 27]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 28]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 29]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				
0 0 0 0 0 0 0				
0 2 1 1 1 0 0				
0 1 1 2 2 0 0				
0 0 0 0 0 0 0				
$x[:, :, 30]$				
0 0 0 0 0 0 0				
0 1 0 0 0 1 0				

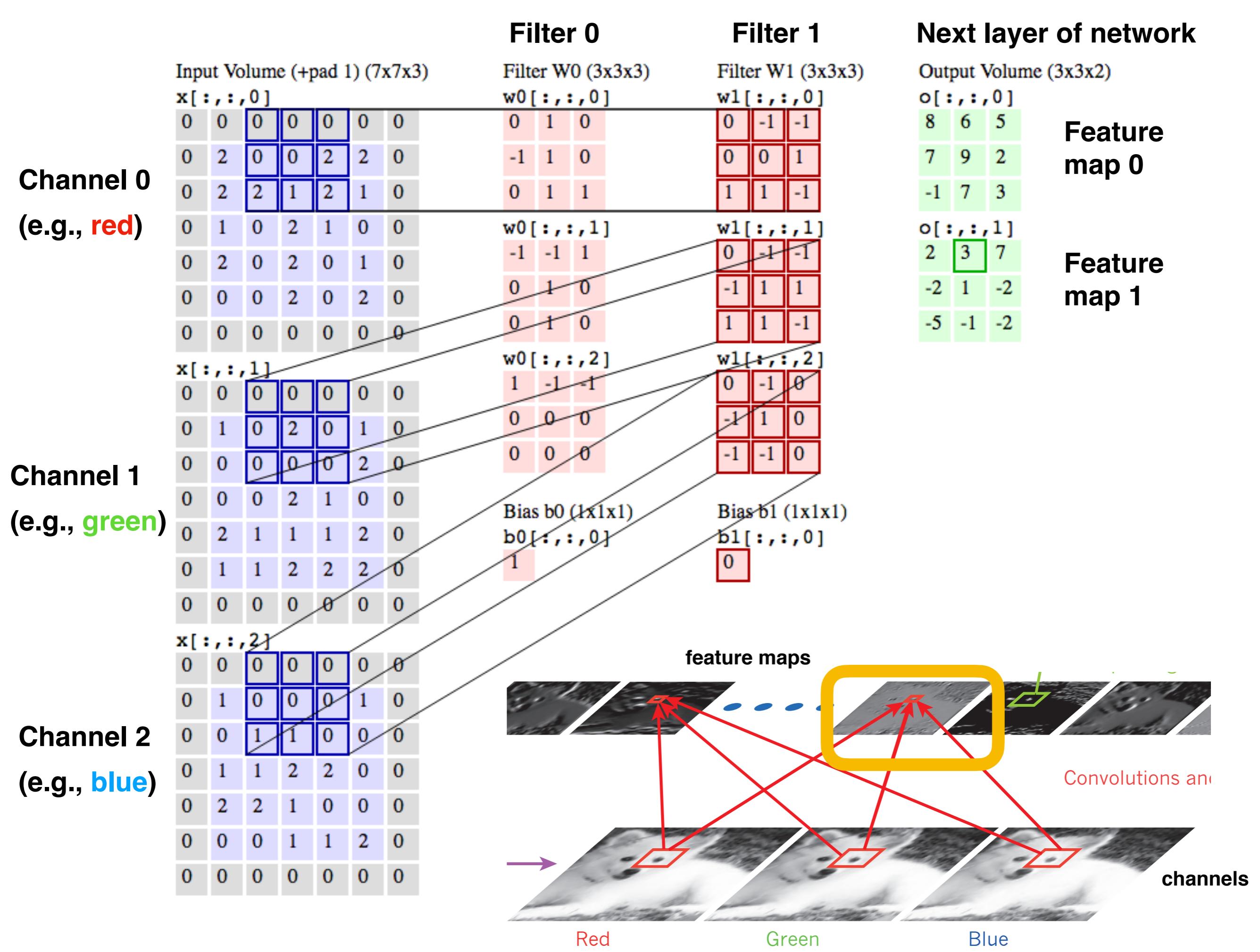
Let's skip to compute the next feature map...



Channel 0
(e.g., red)

Input Volume (+pad 1) (7x7x3)							Filter 0			Filter 1			Next layer of network			
$x[:, :, 0]$							$w_0[:, :, 0]$			$w_1[:, :, 0]$			$o[:, :, 0]$			
0	0	0	0	0	0	0	0	1	0	0	-1	-1	8	6	5	
0	2	0	0	2	2	0	-1	1	0	0	0	1	7	9	2	
0	2	2	1	2	1	0	0	1	1	1	1	-1	-1	7	9	2
0	1	0	2	1	0	0	$w_0[:, :, 1]$			$w_1[:, :, 1]$			$o[:, :, 1]$			
0	2	0	2	0	1	0	-1	-1	1	0	-1	1	2	3	7	
0	0	0	2	0	2	0	0	1	0	-1	1	1	-2	1	-2	
0	0	0	0	0	0	0	0	1	0	1	1	-1	-5	-1	-2	
$x[:, :, 1]$							$w_0[:, :, 2]$			$w_1[:, :, 2]$			$o[:, :, 1]$			
0	0	0	0	0	0	0	1	-1	-1	0	-1	0	2	3	7	
0	1	0	2	0	1	0	0	0	0	-1	1	0	-2	1	-2	
0	0	0	0	0	2	0	0	0	0	-1	-1	0	-5	-1	-2	
0	0	0	2	1	0	0	$b_0[:, :, 0]$			$b_1[:, :, 0]$			$o[:, :, 2]$			
0	2	1	1	1	2	0	1	-1	-1	0	-1	0	2	3	7	
0	1	1	2	2	2	0	0	0	0	-1	1	0	-2	1	-2	
0	0	0	0	0	0	0	0	0	0	-1	-1	0	-5	-1	-2	
$x[:, :, 2]$							Bias b_0 (1x1x1)			Bias b_1 (1x1x1)			$o[:, :, 2]$			
0	0	0	0	0	0	0	1			0			2	3	7	
0	1	0	0	0	1	0	0						-2	1	-2	
0	0	1	0	1	0	0	0						-5	-1	-2	



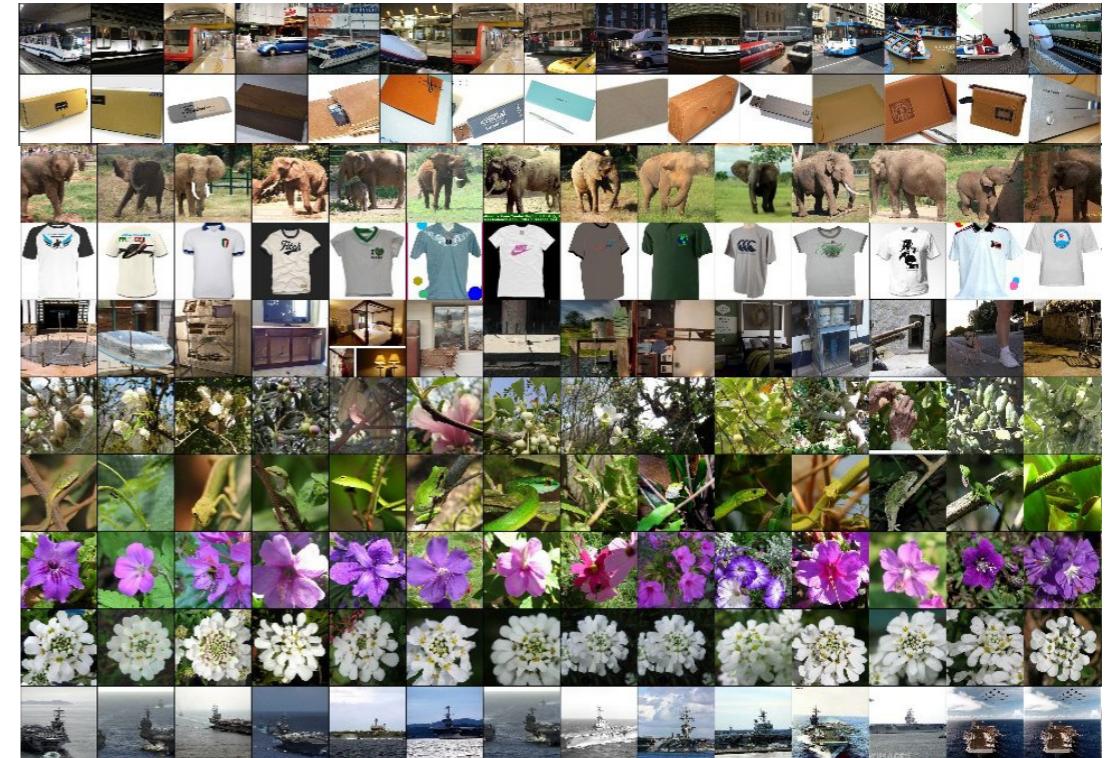


ConvNets are trained via gradient descent

ConvNets are trained on a lot of data

Training data (ImageNet)

- Usually trained on ImageNet
- 1.2 million images with labels
- 1000 categories



These models have millions of learnable parameters

AlexNet

8 layers

~60 million parameters

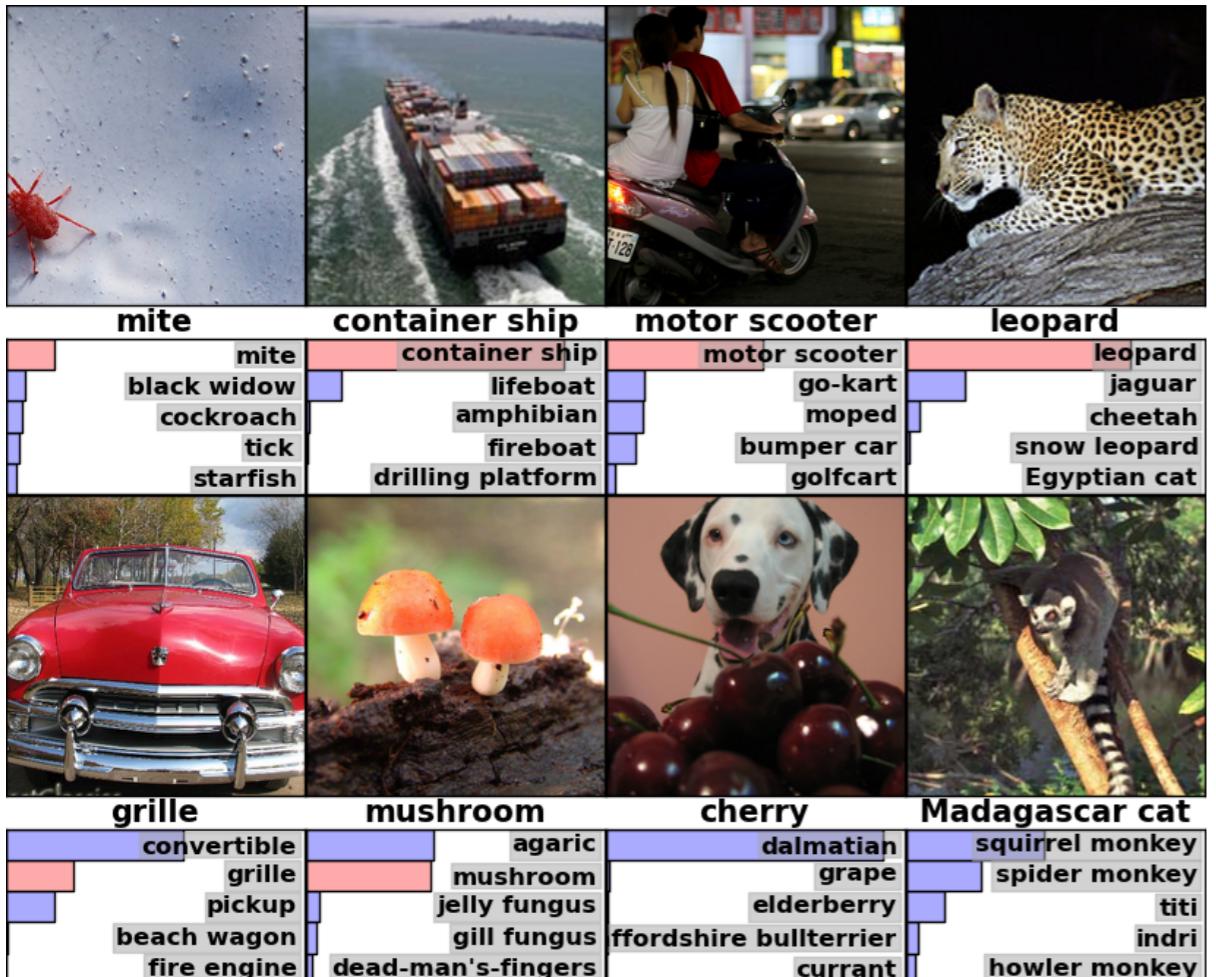
GoogLeNet

22 layers

~5 million parameters

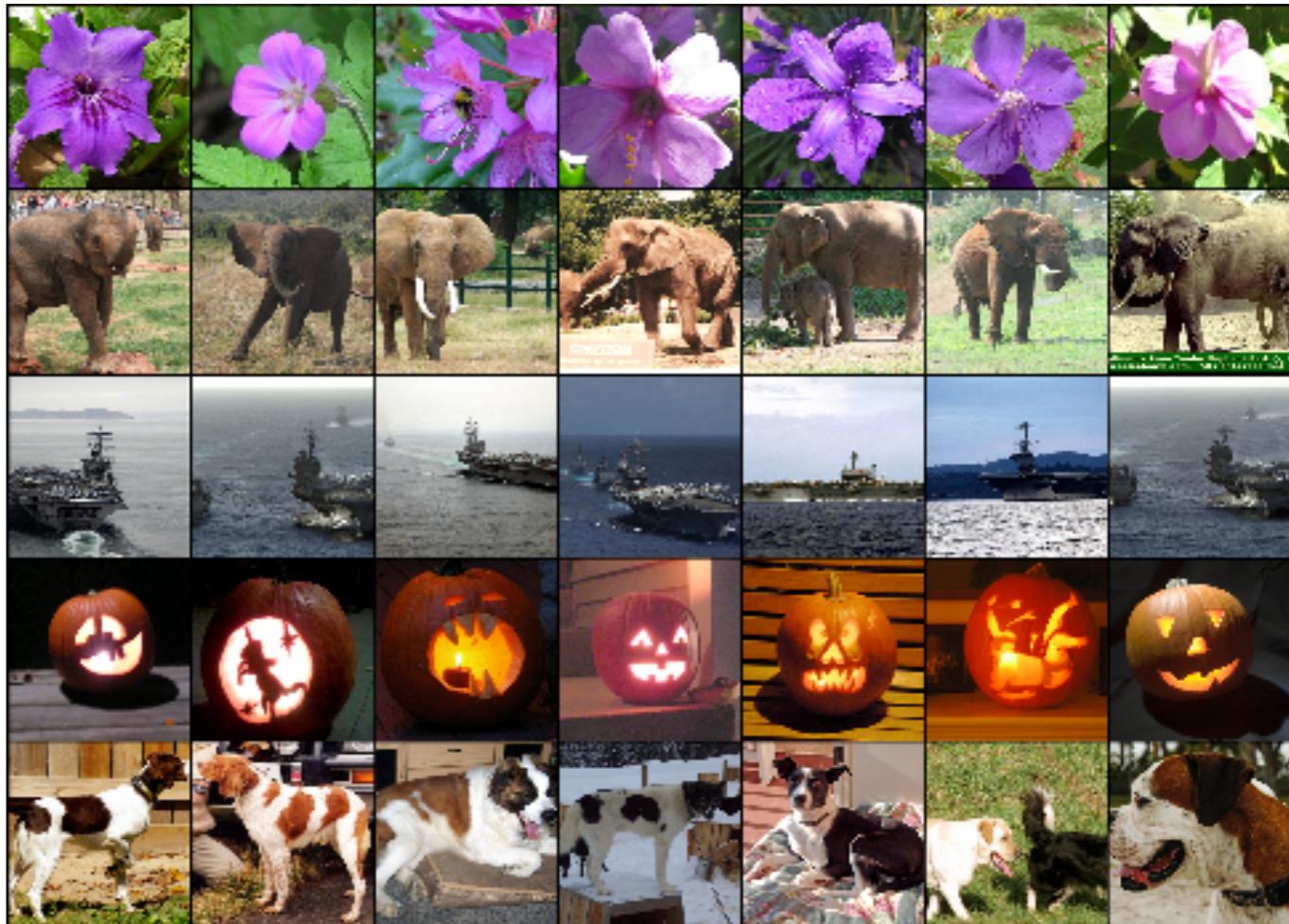
Example results on test images

object recognition



finding similar images

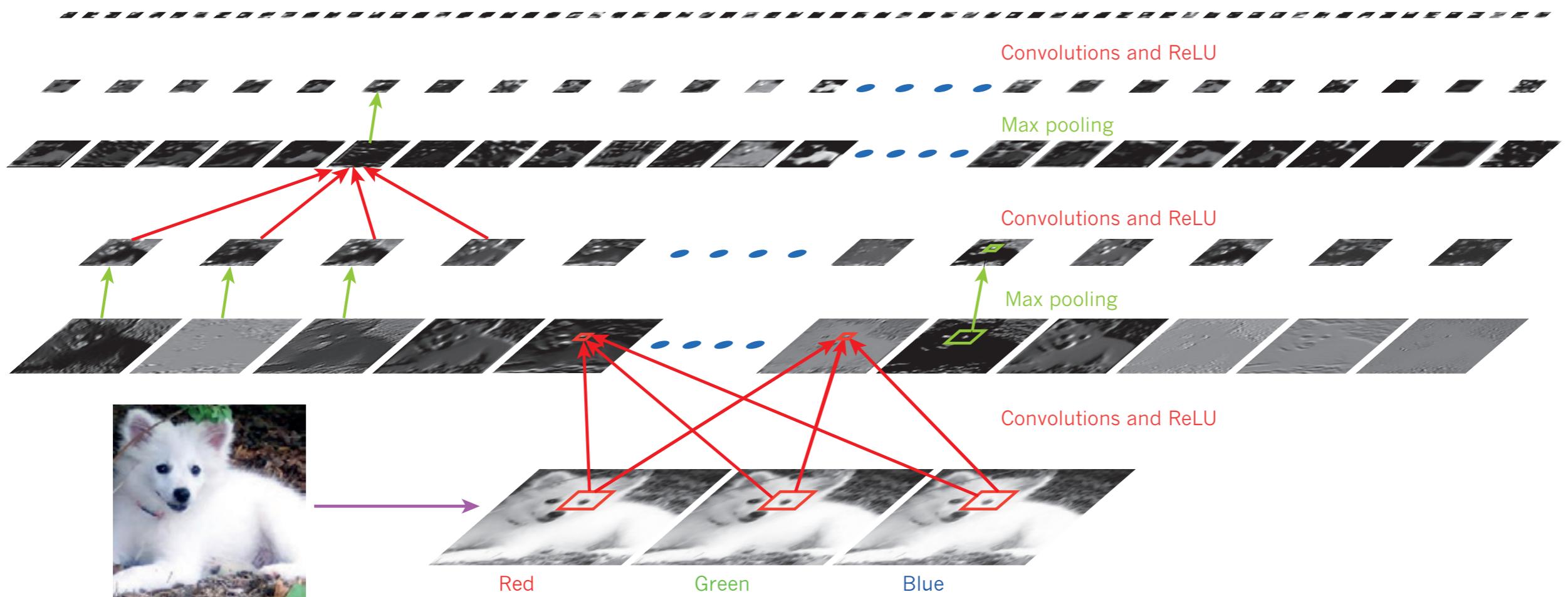
probe similar images...



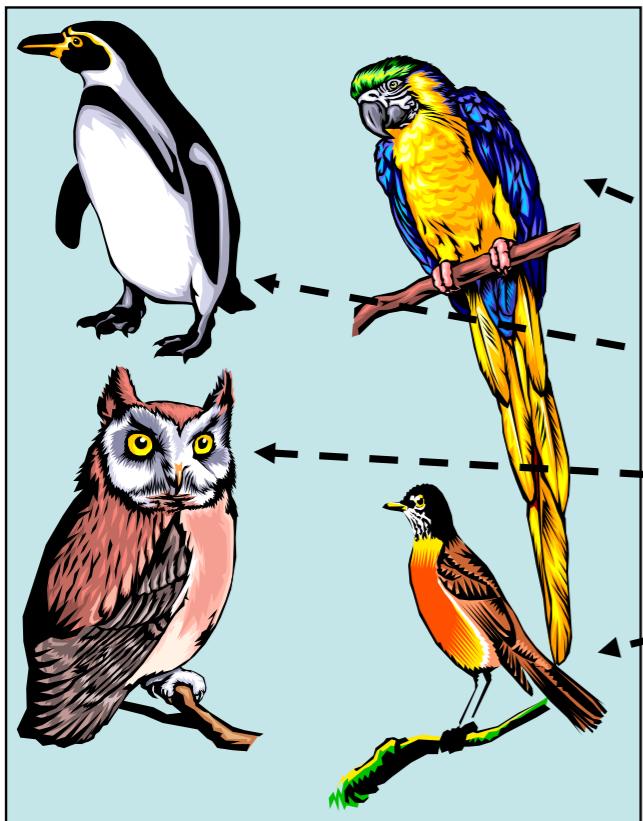
Is it a prototype model? Is it an exemplar model?

- well, it's not really either type of model. nor is it a mixture model like the rational model.
- It could choose to use its millions of parameters to store individual examples, abstract prototypes, or learn clusters

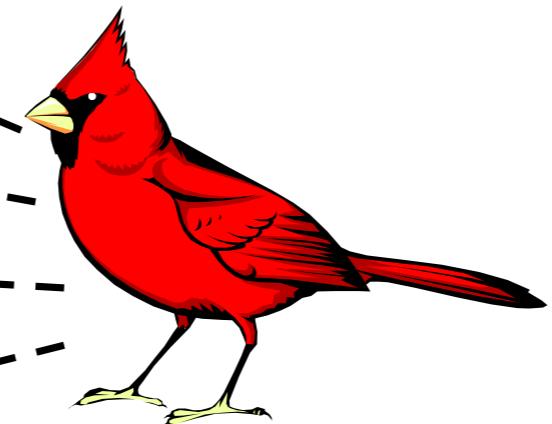
Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)



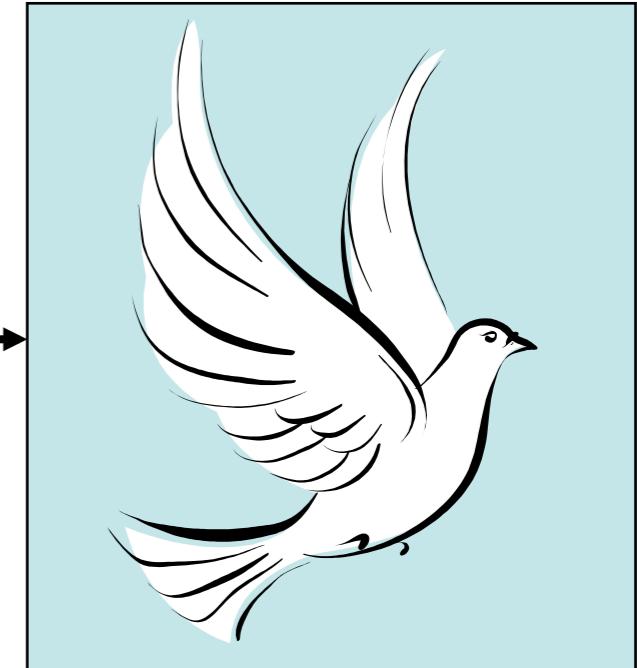
Review: Similar in what respect? What counts as a feature?



Birds
You've Seen



Bird?



Prototypical
Bird

- What counts as a feature? (Murphy & Medin, 1985)
 - To change the importance of age, we could include features for "around 10 years ago," "around 100 years ago," "1000 years ago", etc.
 - To change importance of size, we could include "smaller than the earth," "smaller than a country", "smaller than a city," etc.
- AlexNet doesn't seem to suffer from this philosophical issue – it learns the features it needs to solve the classification problem.

Typicality ratings are highly reliable and predict many aspects of categorization.

Category: furniture

Member	Rank	Score (1 [typical]... 7 [not typical])
chair	1.5	1.04
sofa	1.5	1.04
couch	3.5	1.10
table	3.5	1.10
easy chair	5	1.33
dresser	6.5	1.37
rocking chair	6.5	1.37
coffee table	8	1.38
rocker	9	1.42
love seat	10	1.44
...		
stove	50	5.40
counter	51	5.44
clock	52	5.48
drapes	53	5.67
refrigerator	54	5.70
picture	55	5.75
closet	56	5.95
vase	57	6.23
ashtray	58	6.35
fan	59	6.49
telephone	60	6.68

Instructions:

...‘Which breeds of dogs are real “doggy dogs”? To me, a German Shepard is very doggy, but a Pekinese is not. “Rate the extent to which each instance represents your idea of image of the meaning of the category”

Goodness ratings are **highly consistent** across participant

(Rosch, 1975)

Can you use ConvNets to predict category typicality?

Lake, B. M., Zaremba, W., Fergus, R., & Gureckis, T. M. (2015). Deep Neural Networks Predict Category Typicality Ratings for Images.



typical dog



atypical dog

- For people, typicality influences performance in practically any category-related task
 - speed of categorization
 - ease of production
 - ease of learning
 - usefulness for inductive inference
- No known model successfully predicts typicality ratings from raw images -- How do convnets perform?

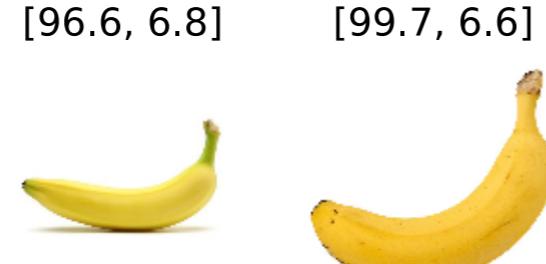
Category: Banana ($\rho=0.82$)

How well does this picture fit your idea or image of the category? (rated on 1-7 scale)

Human typicality ratings

Most typical →

[97.8, 6.8] [98.0, 6.8] [96.6, 6.8] [99.7, 6.6]



[96.9, 6.6]

[99.3, 6.0]

[78.6, 5.8]

[99.5, 5.5]



[12.1, 5.3]

[59.7, 4.4]

[2.9, 4.3]

[46.1, 4.1]



[14.0, 4.1]

[0.2, 3.6]

[2.3, 2.5]

[1.3, 2.4]



Least typical

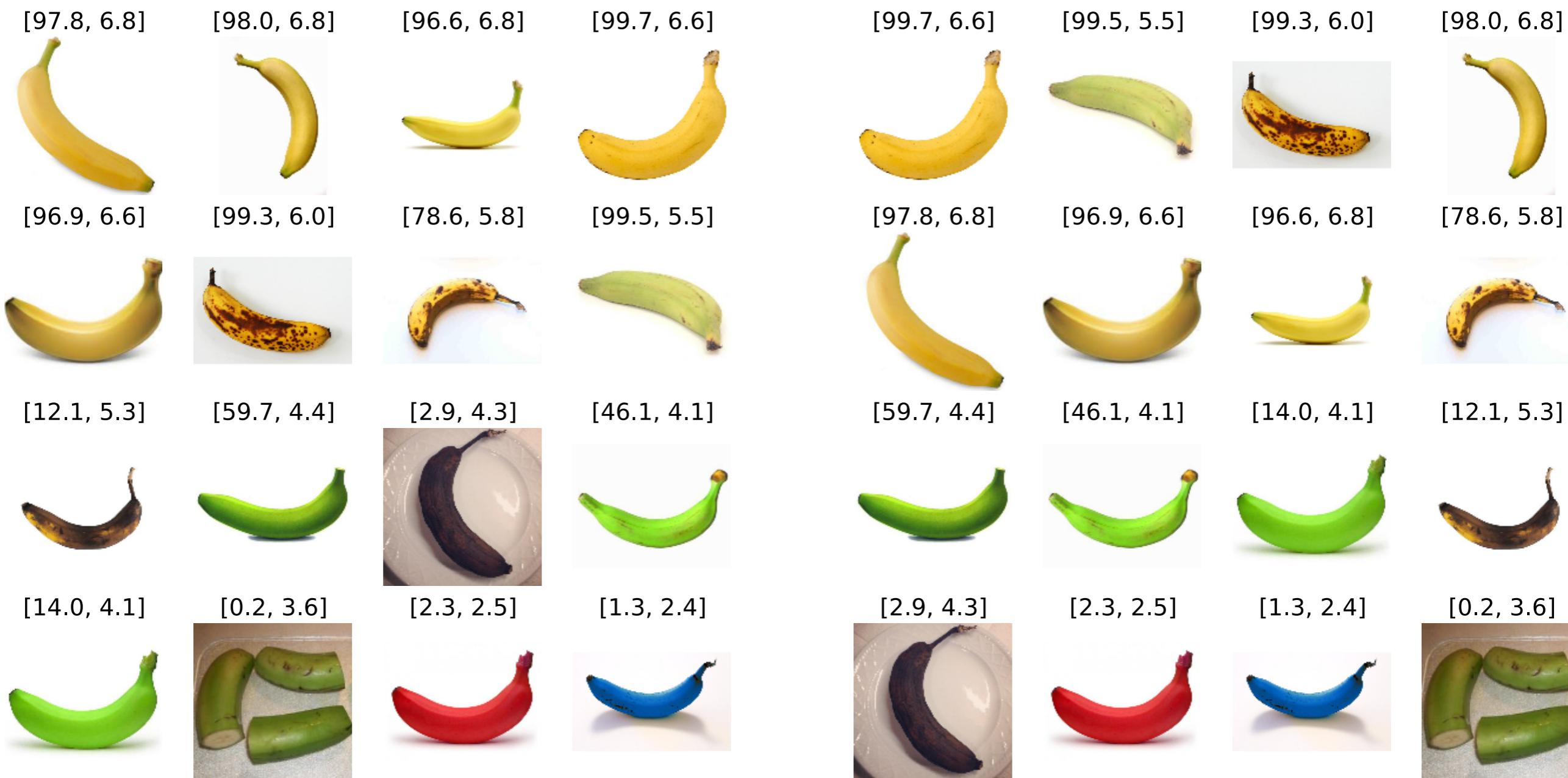
rating key: [machine (0-100), human (1-7)]

Category: Banana ($\rho=0.82$)

How well does this picture fit your idea or image of the category? (rated on 1-7 scale)

Human typicality ratings

Most typical →



Convnet typicality ratings

rating key: [machine (0-100), human (1-7)]

Category: Bathtub ($\rho=0.68$)

Human typicality ratings

Most typical



[60.6, 6.6]



[72.0, 6.1]



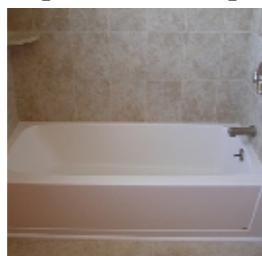
[67.6, 5.6]



[1.0, 3.0]



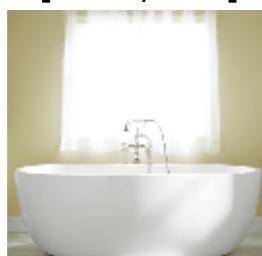
[58.5, 6.6]



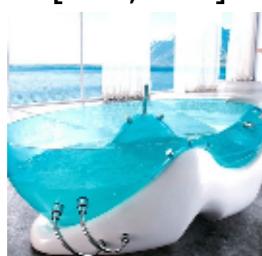
[80.7, 6.0]



[63.0, 5.2]



[1.5, 2.9]



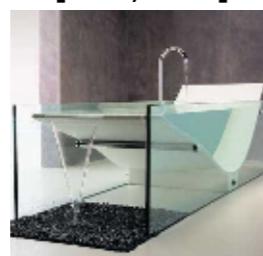
[57.3, 6.6]



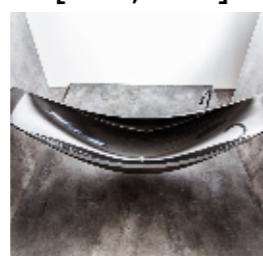
[9.5, 5.9]



[9.8, 3.2]



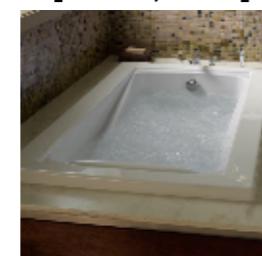
[1.0, 2.8]



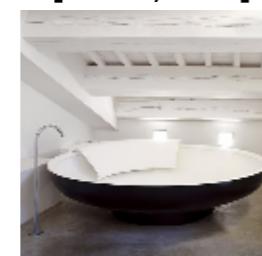
[66.5, 6.2]



[35.4, 5.7]



[16.4, 3.1]



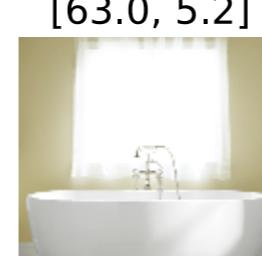
[9.1, 2.4]



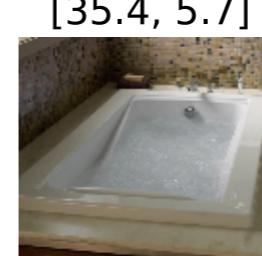
[80.7, 6.0]



[63.0, 5.2]



[35.4, 5.7]

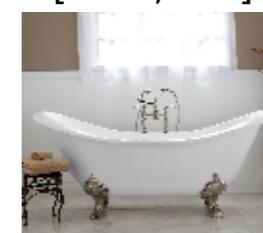


[9.1, 2.4]



Convnet typicality ratings

[72.0, 6.1]



[60.6, 6.6]



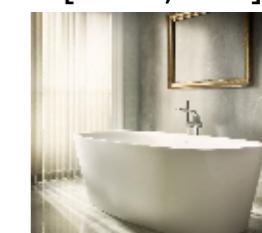
[16.4, 3.1]



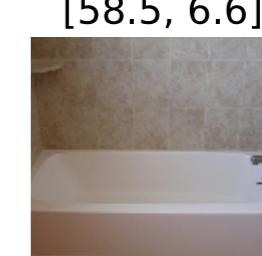
[1.5, 2.9]



[67.6, 5.6]



[58.5, 6.6]



[9.8, 3.2]



[1.0, 2.8]



[66.5, 6.2]



[57.3, 6.6]



[9.5, 5.9]



[1.0, 3.0]



Least typical

rating key: [machine (0-100), human (1-7)]

Category: Envelope ($\rho=0.79$)

Human typicality ratings

Most typical



[91.5, 6.7]



[75.2, 6.6]



[96.4, 6.6]



[98.5, 6.6]



[98.5, 6.6]



[97.7, 6.6]



[96.4, 6.6]



[91.5, 6.7]



[97.7, 6.6]



[82.8, 6.2]



[69.5, 5.3]



[59.7, 5.2]



[82.8, 6.2]



[75.2, 6.6]



[69.5, 5.3]



[59.7, 5.2]



[31.2, 5.1]



[32.5, 5.1]



[10.8, 4.8]



[5.8, 4.2]



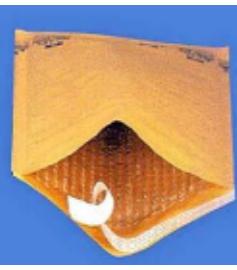
[50.6, 3.8]



[41.9, 3.4]



[32.5, 5.1]



[31.2, 5.1]



[10.4, 4.1]



[50.6, 3.8]



[41.9, 3.4]



[24.9, 3.2]



[24.9, 3.2]



[10.8, 4.8]



[10.4, 4.1]



[5.8, 4.2]



Least typical

Convnet typicality ratings

[98.5, 6.6]



[97.7, 6.6]



[96.4, 6.6]



[91.5, 6.7]



[82.8, 6.2]



[75.2, 6.6]



[69.5, 5.3]



[59.7, 5.2]



[50.6, 3.8]



[41.9, 3.4]



[32.5, 5.1]



[31.2, 5.1]



[24.9, 3.2]



[10.8, 4.8]



[10.4, 4.1]



[5.8, 4.2]



rating key: [machine (0-100), human (1-7)]

Category: Teapots ($\rho=0.38$)

Human typicality ratings

Most typical →

[95.8, 6.6] [98.8, 6.6] [93.5, 6.4] [98.1, 6.2]



[46.0, 6.0] [63.6, 5.8] [95.0, 5.8] [52.8, 5.8]



[97.2, 5.6] [8.4, 5.3] [93.4, 5.2] [34.9, 4.9]



[78.8, 4.8] [98.5, 4.6] [8.9, 4.3] [83.9, 3.4]



Least typical

Convnet typicality ratings

[98.8, 6.6] [98.5, 4.6] [98.1, 6.2] [97.2, 5.6]



[95.8, 6.6] [95.0, 5.8] [93.5, 6.4] [93.4, 5.2]



[83.9, 3.4] [78.8, 4.8] [63.6, 5.8] [52.8, 5.8]



[46.0, 6.0] [34.9, 4.9] [8.9, 4.3] [8.4, 5.3]

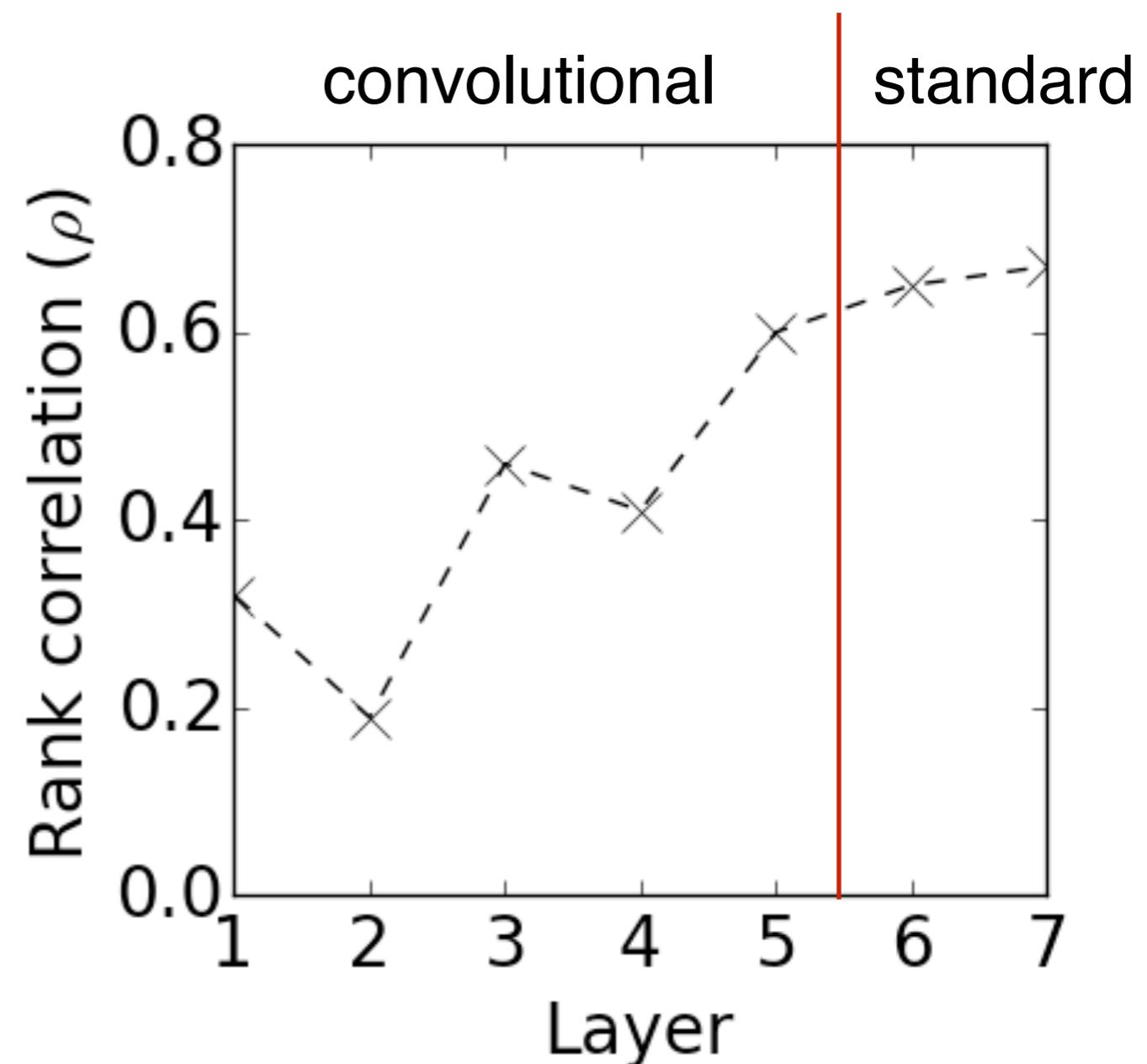


rating key: [machine (0-100), human (1-7)]

Summary of typicality predictions

	Rank Correlation
Banana	0.82
Bathtub	0.68
Coffee Mug	0.62
Envelope	0.79
Pillow	0.67
Soap dispenser	0.74
Table lamp	0.69
Teapot	0.38
Average	0.67

Prediction quality varies as a function of network depth.



Can a “Deep-ALCOVE” predict SHJ results?

Modeling artificial category learning from pixels: Revisiting Shepard, Hovland, and Jenkins (1961) with deep neural networks

Alexa R. Tartaglini (art481@nyu.edu)
Wai Keen Vong (waikeen.vong@nyu.edu)
Brenden M. Lake (brenden@nyu.edu)

Center for Data Science, 60 5th Ave,
New York University, NY, 10011

Abstract

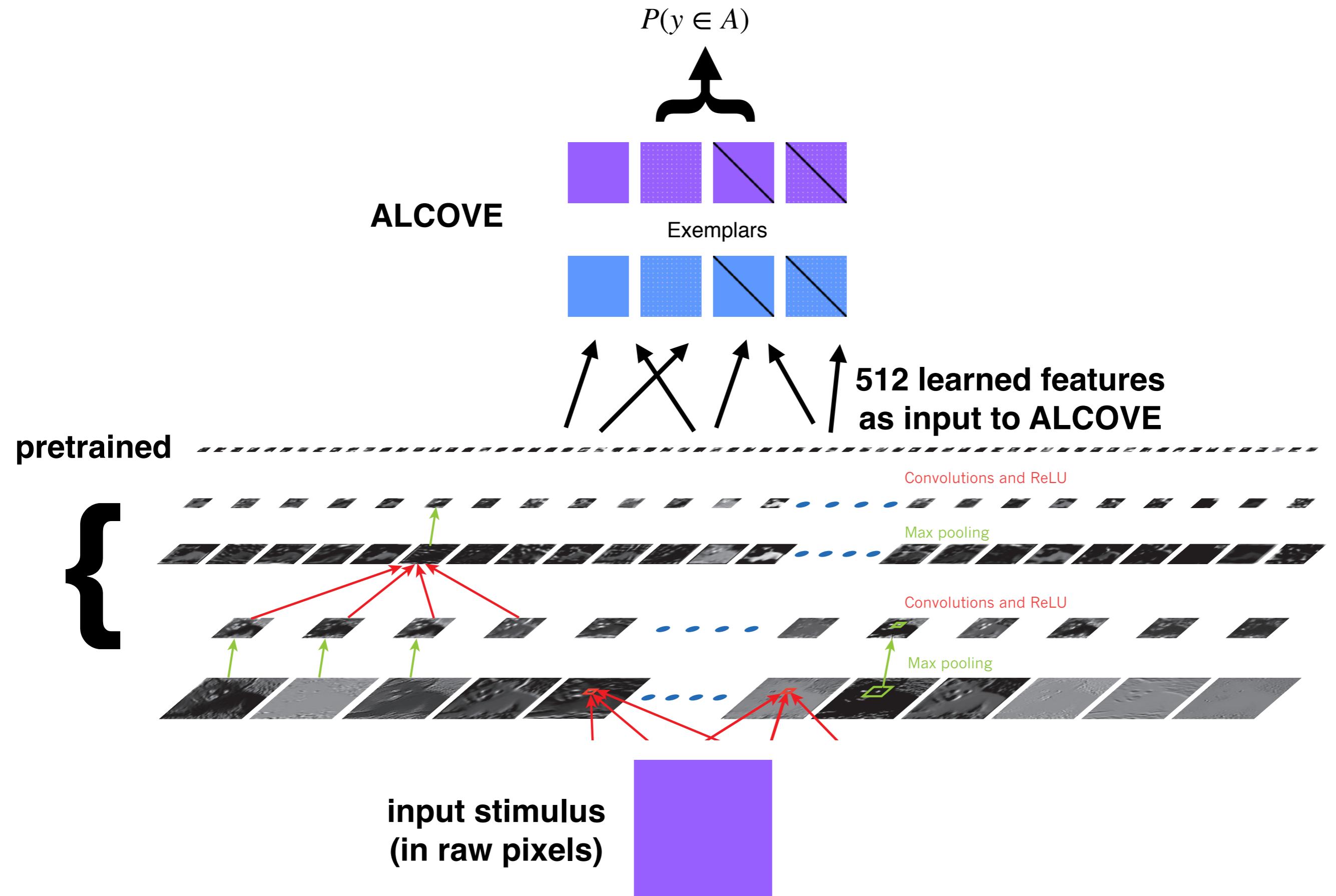
Recent work has paired classic category learning models with convolutional neural networks (CNNs), allowing researchers to study categorization behavior from raw image inputs. However, this research typically uses naturalistic images, which assess participant responses to existing categories; yet, much of traditional category learning research has focused on using novel, artificial stimuli to examine the learning process behind how people acquire categories. In this work, we pair a CNN with ALCOVE (Kruschke, 1992), a well-known exemplar model of categorization, and attempt to examine whether this model can reproduce the classic type ordering effect from Shepard, Hovland, and Jenkins (1961) on raw images rather than abstract features. We examine this question with a variety of CNN architectures and image datasets and compare ALCOVE-CNN to two other models that lacked certain key features of ALCOVE. We found that our ALCOVE-CNN model could reproduce the type ordering effect more often than the other models we tested, but in limited situations. Our results showed that success varied greatly across the various configurations we tested, suggesting that the feature representations from CNNs provide strong constraints in properly capturing this effect.

Keywords: category learning; convolutional neural networks; exemplar models; attention

and categorization. For instance, CNNs pre-trained for image classification have shown success in predicting category typicality ratings (Lake, Zaremba, Fergus, & Gureckis, 2015) and similarity ratings from natural images (Peterson, Abbott, & Griffiths, 2018). More recently, researchers have begun to combine CNNs with classic prototype and exemplar models of categorization (Battleday, Peterson, & Griffiths, 2020; Guest & Love, 2019; Singh, Peterson, Battleday, & Griffiths, 2020; Nosofsky, Meagher, & Kumar, 2020), usually with the aim of predicting human categorization decisions for images of common categories such as animals and vehicles.

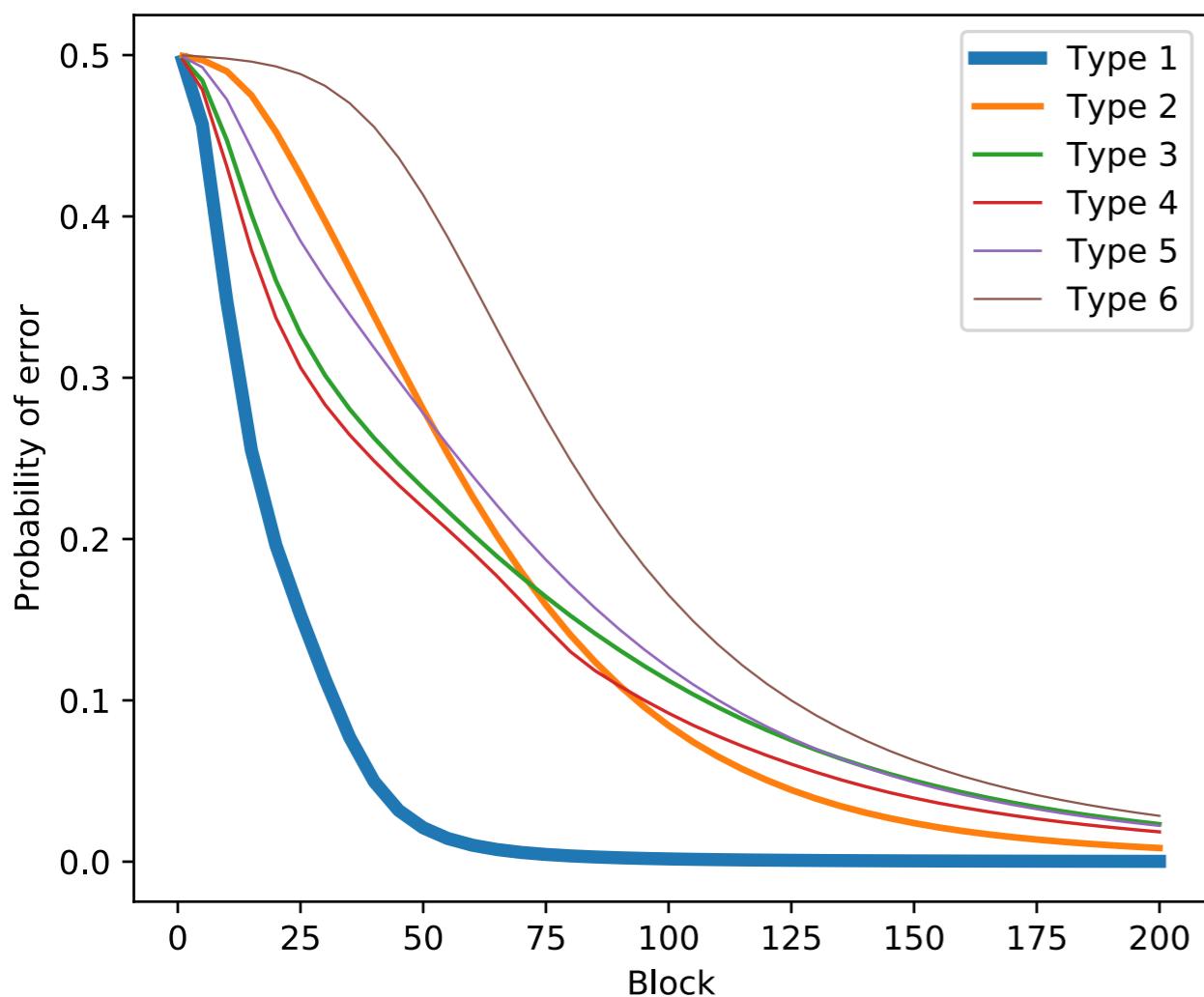
These successes in predicting human judgments on natural images do not, however, imply that such an approach will be a successful psychological model of categorization as studied in its most classic setting: artificial category learning tasks conducted in the lab. This is the question we take up in our work here. One of the most influential findings in this vein is the classic experiment by Shepard et al. (1961) which has been replicated many times (Nosofsky, Gluck, Palmeri, McKinley, & Glauthier, 1994; Crump, McDonnell, & Gureckis, 2013). Shepard et al. (1961) showed that with

Can a “Deep-ALCOVE” predict SHJ results?

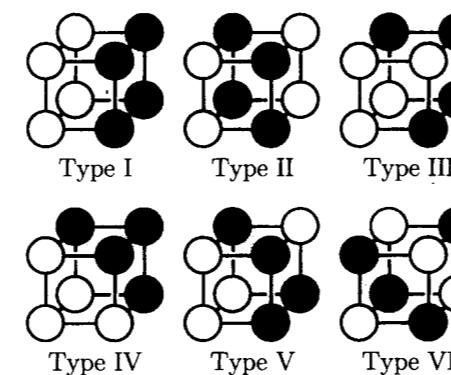
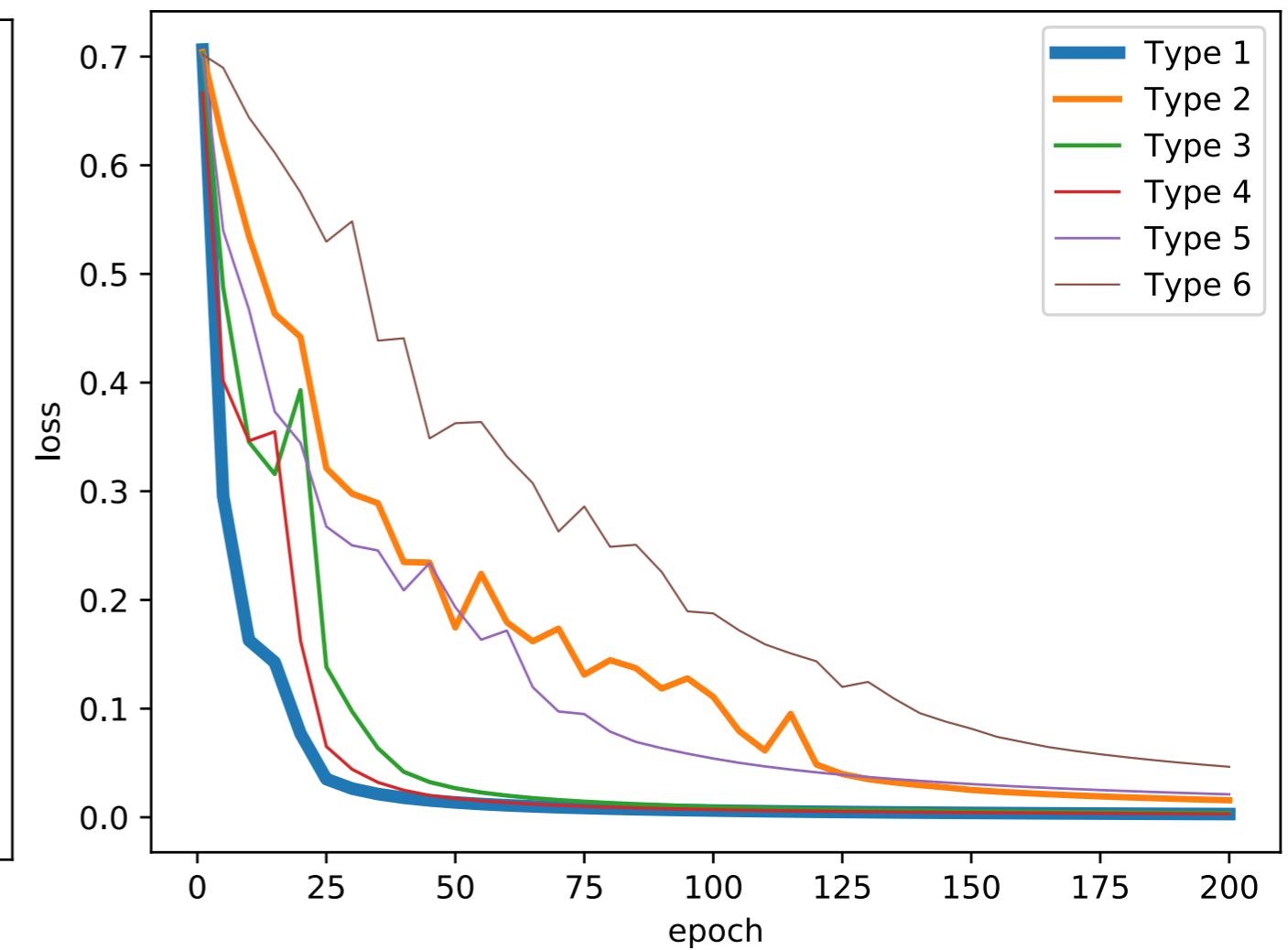


Example ConvNet results on raw SHJ stimuli

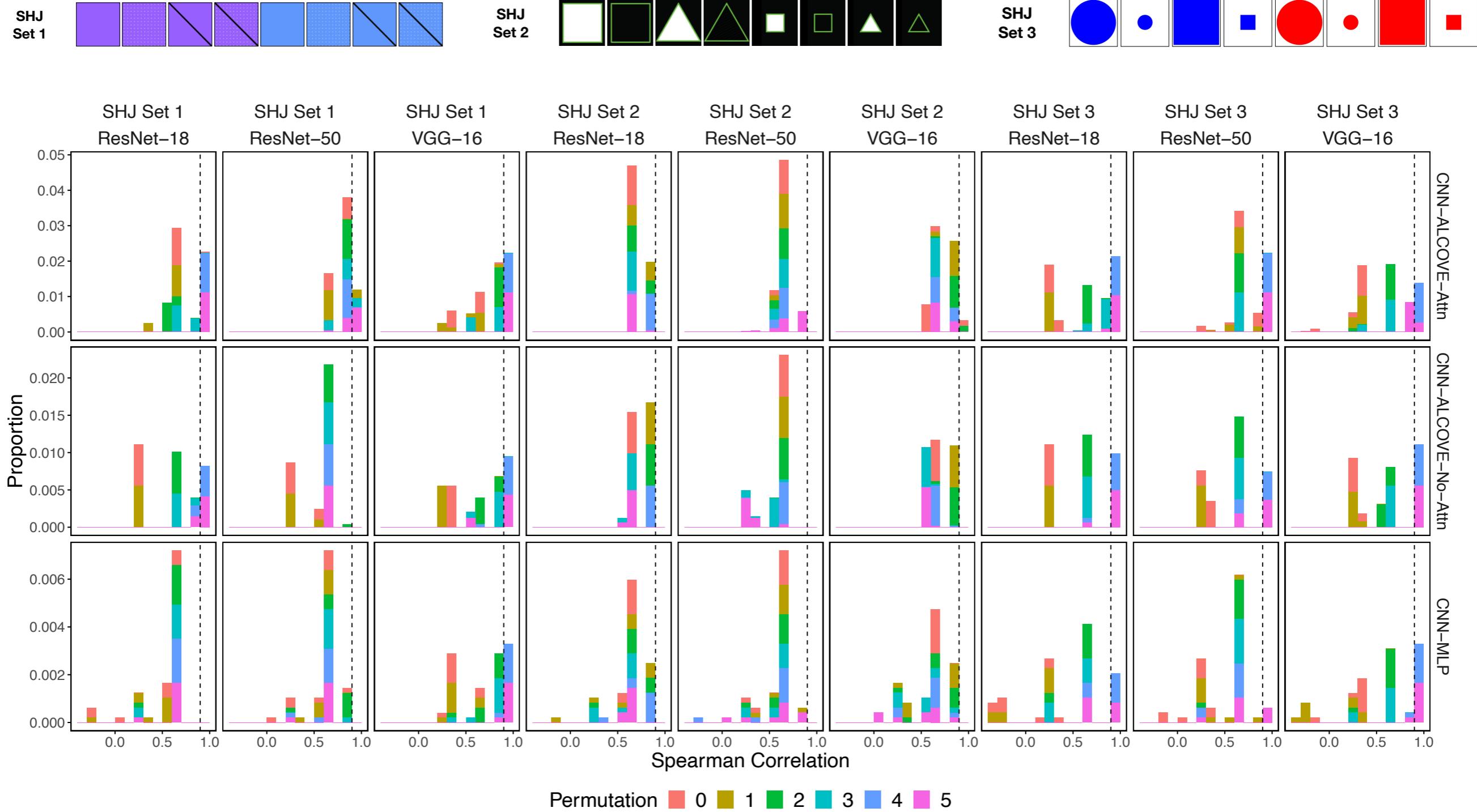
Deep ConvNet + ALCOVE



Deep ConvNet + standard neural net
(gets Type II order wrong)
(gets Type III order wrong)
(gets Type IV order wrong)



But Deep-ALCOVE does not robustly predict the right ordering across different key factors



Some conclusions

- We still don't know exactly what these recent advances in machine learning mean for the cognitive science of concepts
- ConvNets — the dominant algorithm for object classification from images — don't fit neatly into current psychological theories of concepts
- Alexa's results point to difficulties in applying ConvNets out of the box to interface with raw experimental stimuli
- My view is that interactions between these two fields are critical for advancing both

Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)

