Lecture 14: Representation learning

Announcements

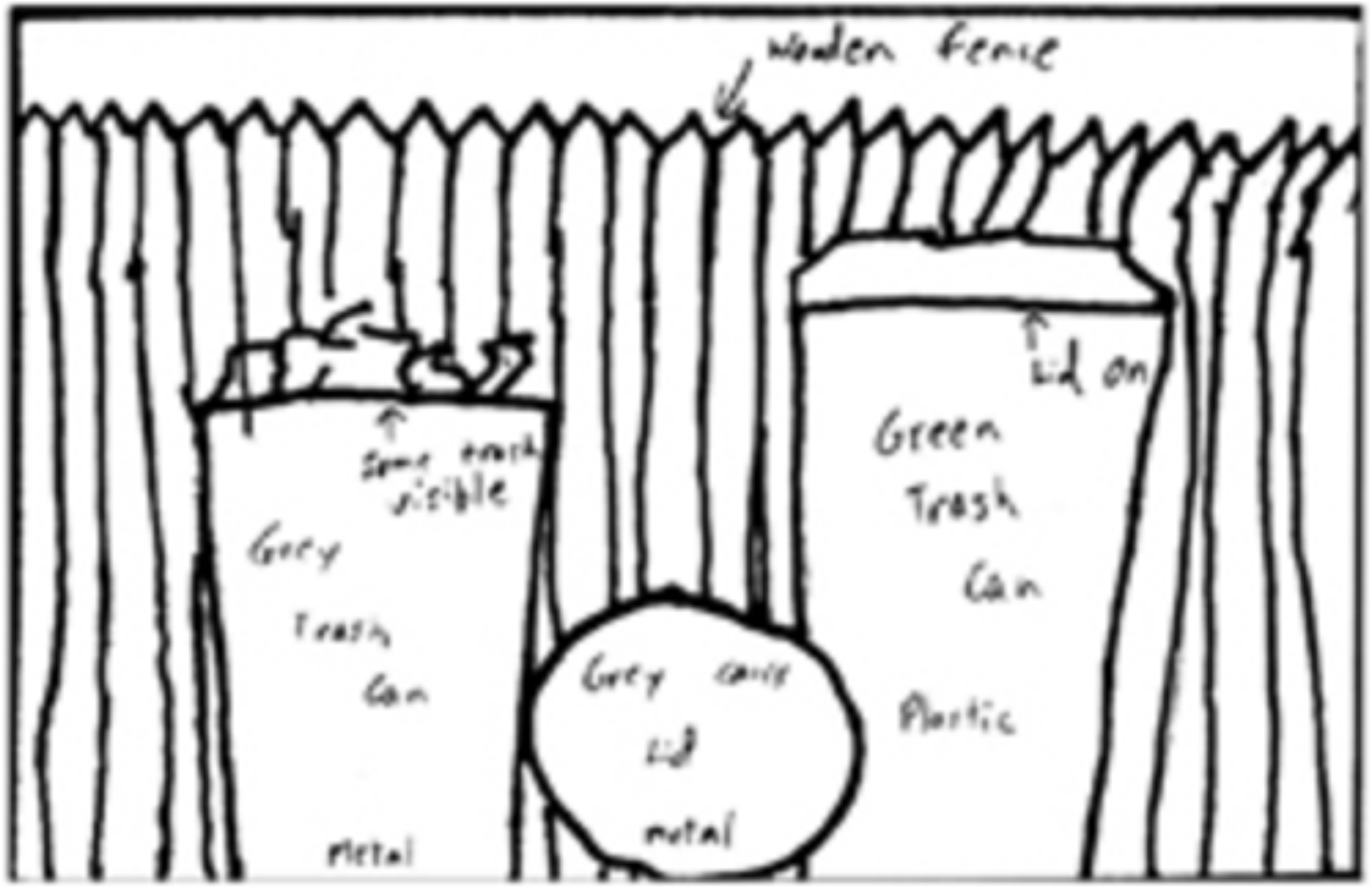
- Project proposal due after spring break
- We'll post description on website
- New idea? Turn in half-page description
- Premade project idea? Just tell us which one

Today

- What representations do neural nets learn?
- Transfer learning
- Unsupervised learning

Observed image

Drawn from memory

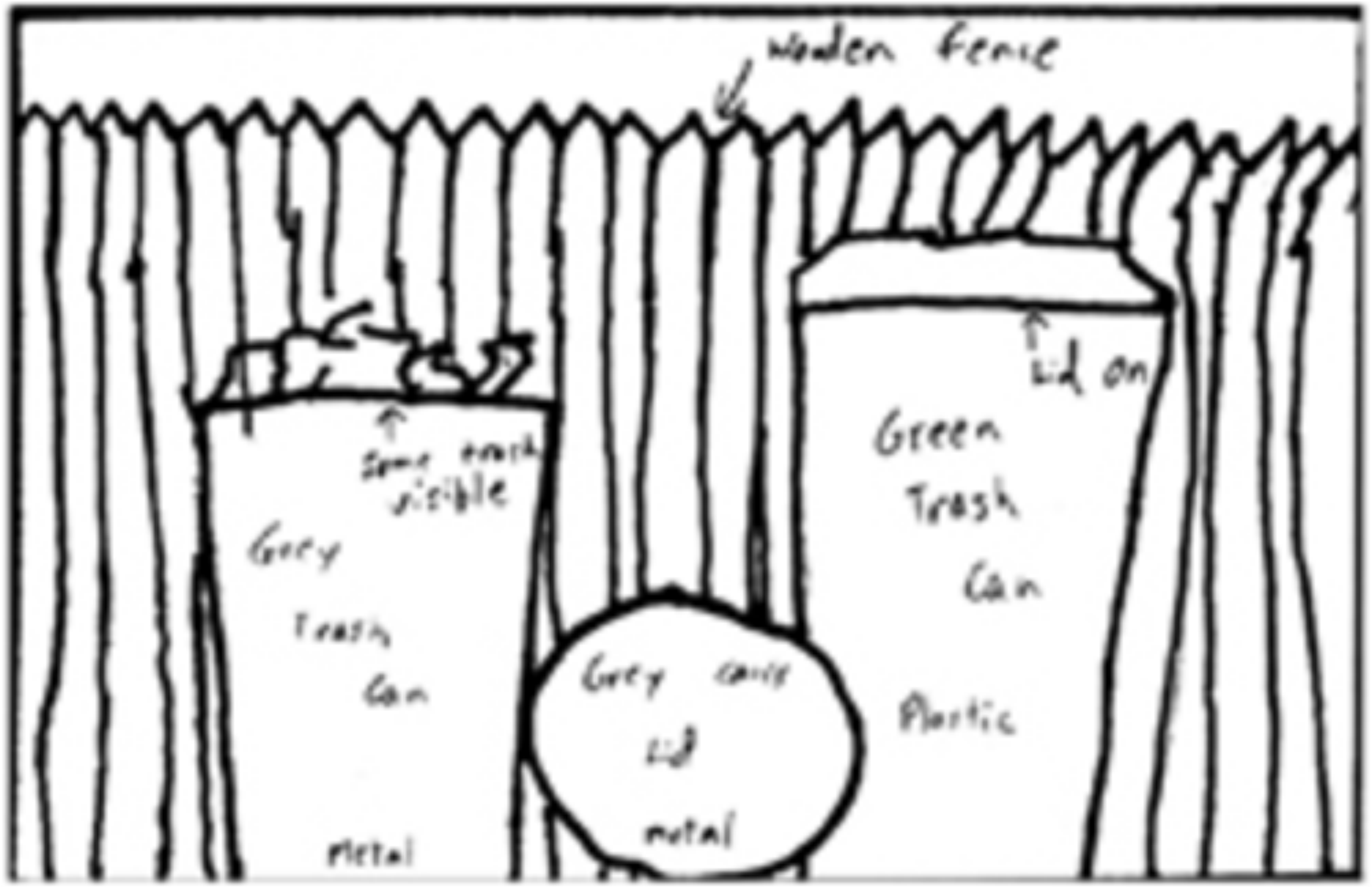


[Bartlett, 1932]
[Intraub & Richardson, 1989]

Source: Isola, Freeman, Torralba

Observed image

Drawn from memory



[Bartlett, 1932]
[Intraub & Richardson, 1989]

Source: Isola, Freeman, Torralba

"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees.”

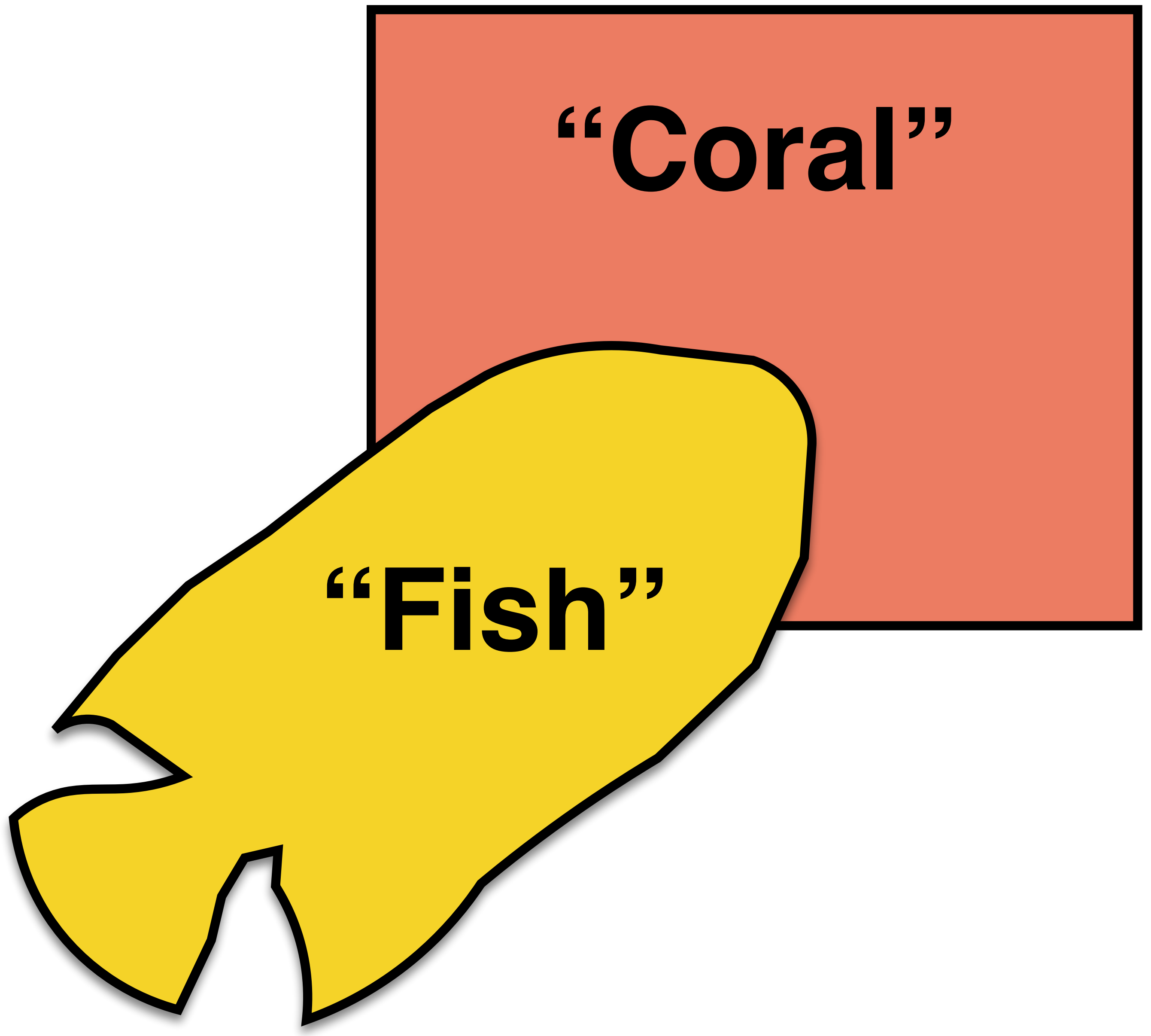
— Max Wertheimer, 1923

Source: Isola, Freeman, Torralba

Representation learning

$\mathbf{x}$

Image



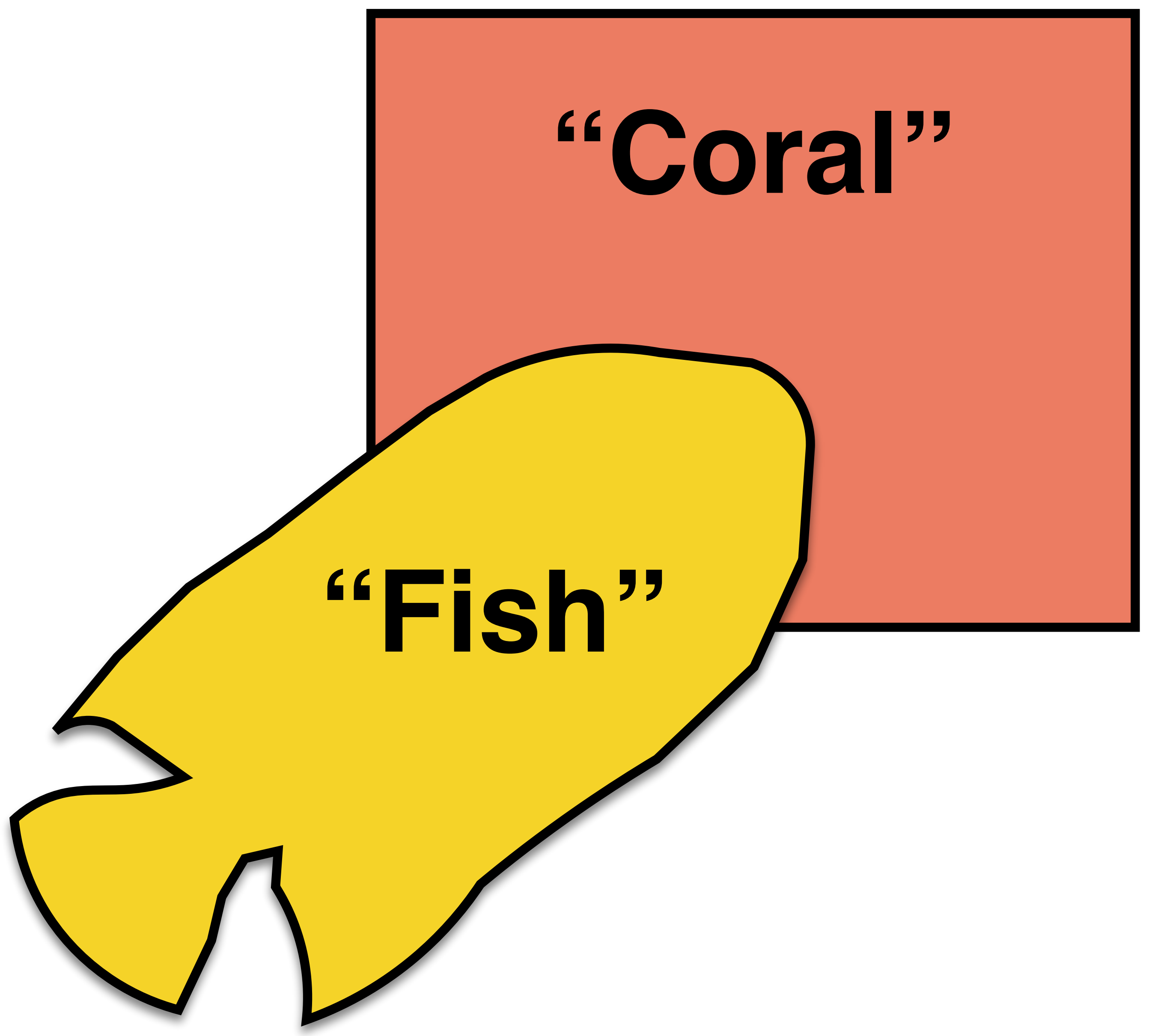
Compact mental representation

Source: Isola, Freeman, Torralba

Representation learning

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Interpretable
5. *Make subsequent problem solving easy*



[See "Representation Learning", Bengio 2013, for more commentary]

Source: Isola, Freeman, Torralba

Transfer learning

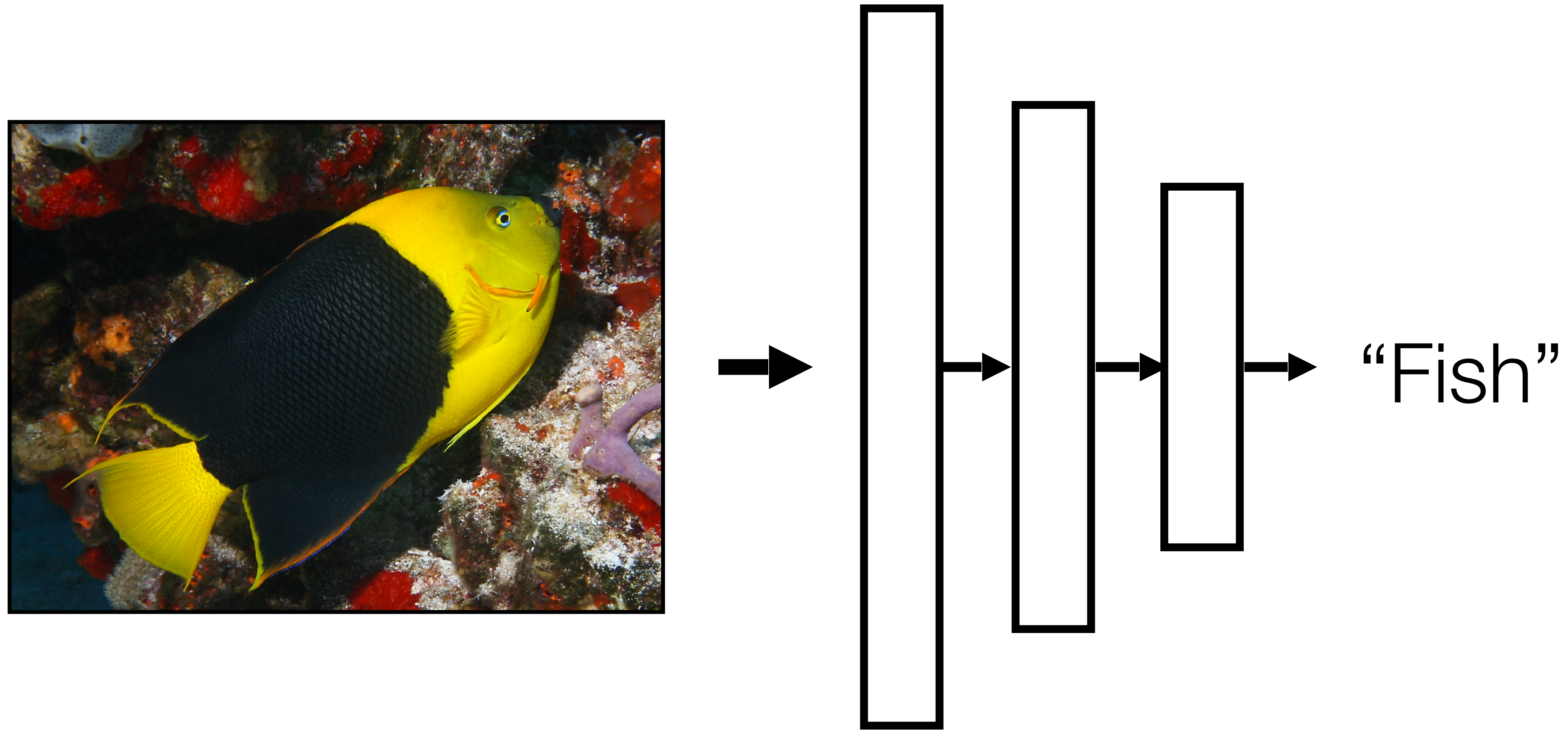
"Generally speaking, a good representation is one that makes a subsequent learning task easier." — *Deep Learning*, Goodfellow et al. 2016

?

Source: Isola, Freeman, Torralba

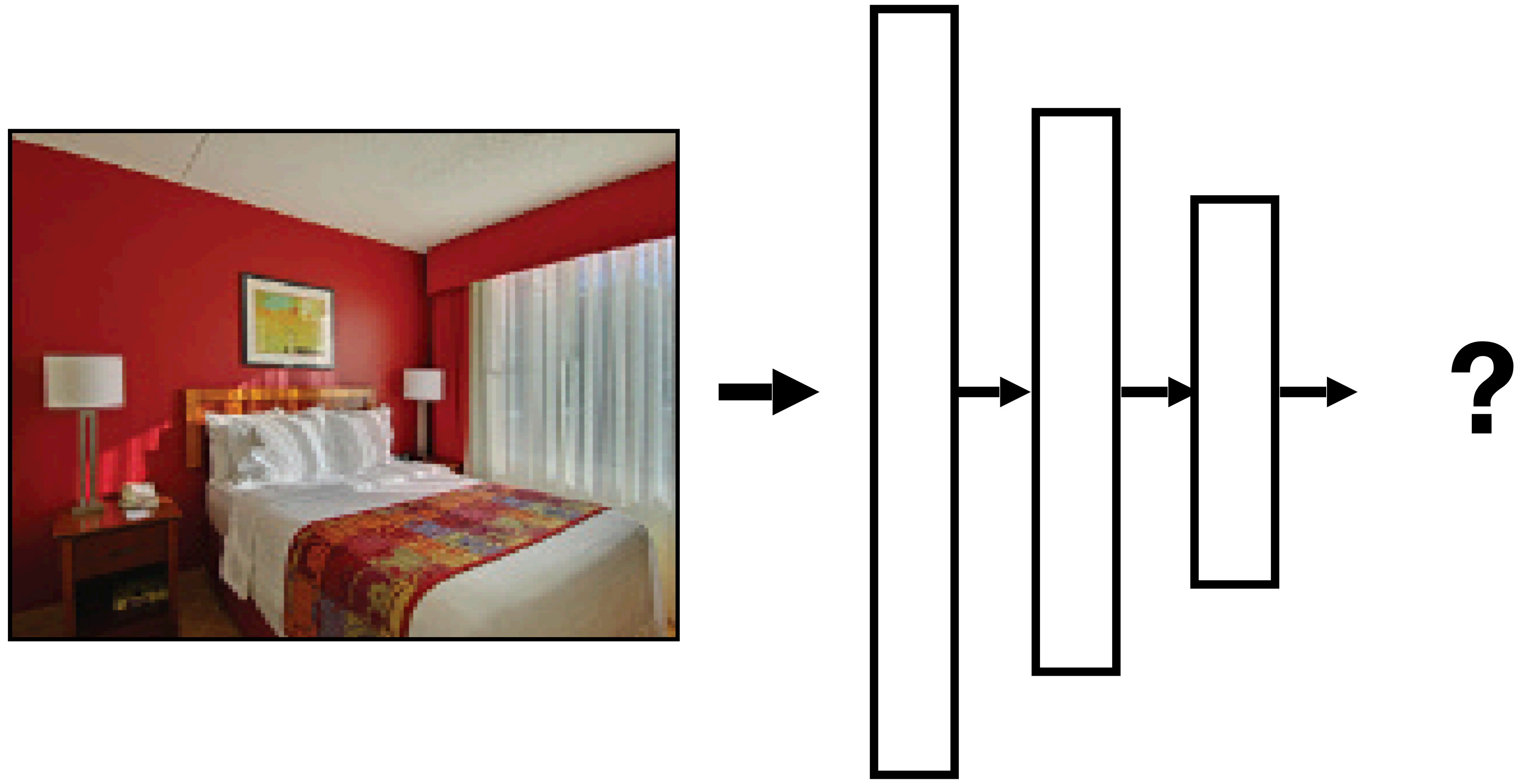
Training

Object recognition



Testing

Place recognition

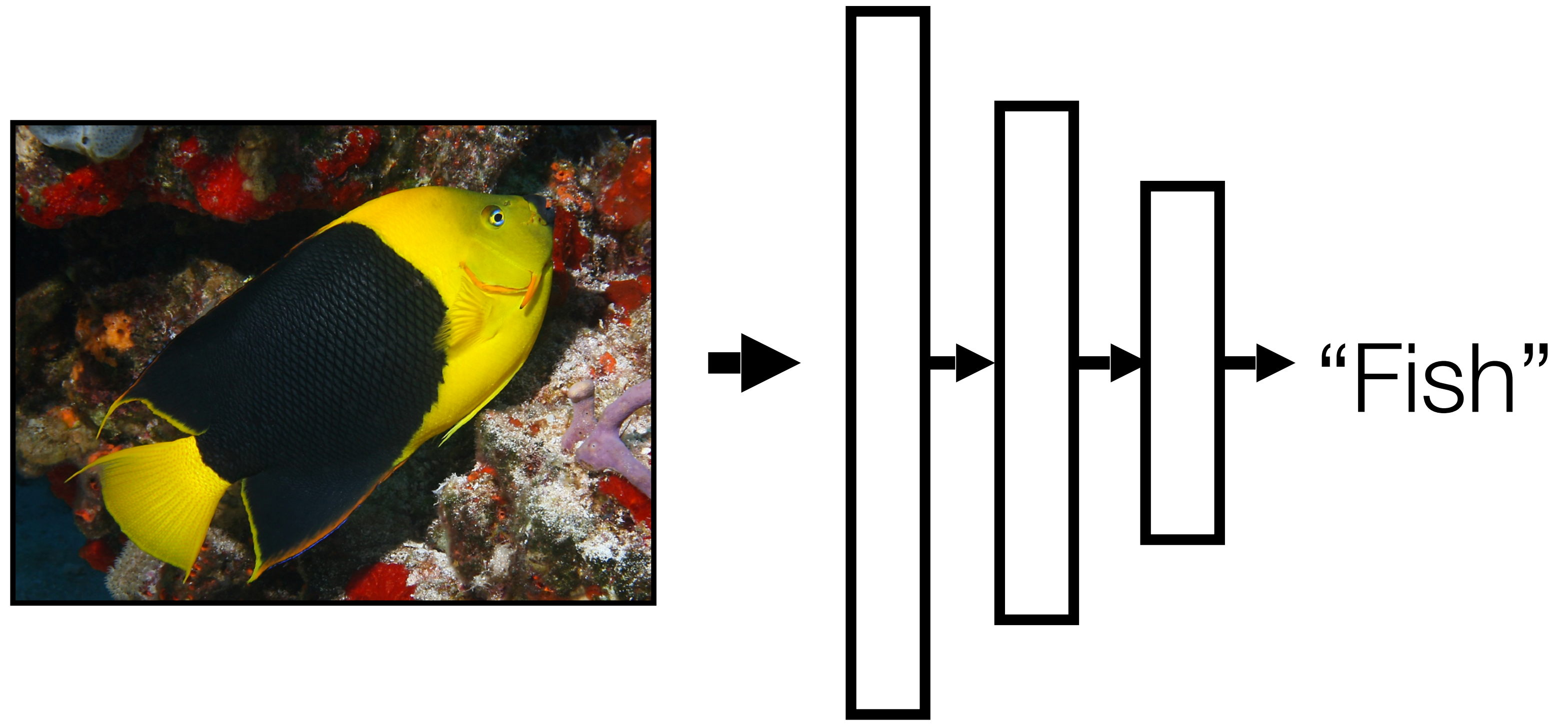


Often, what we will be “tested” on is to learn to do a new thing.

Source: Isola, Freeman, Torralba

Pretraining

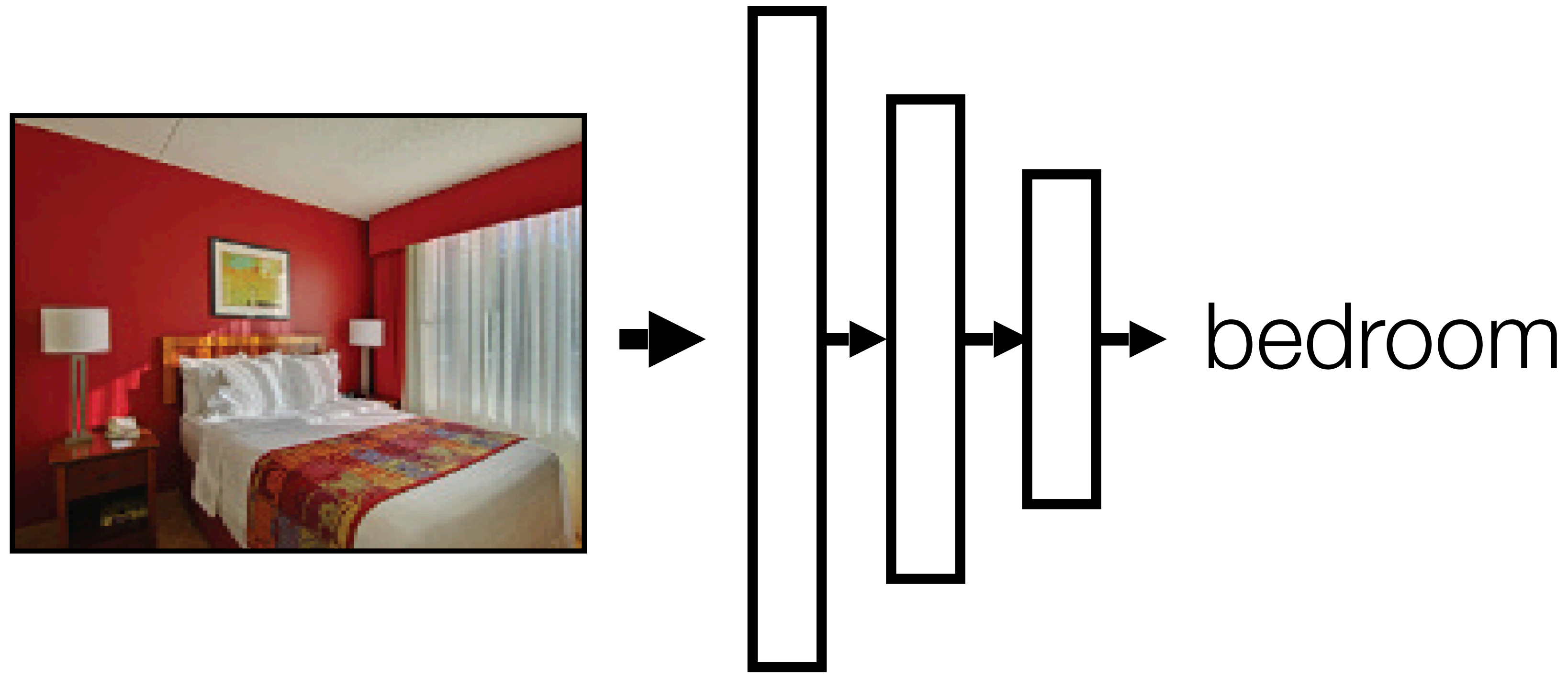
Object recognition



A lot of data

Finetuning

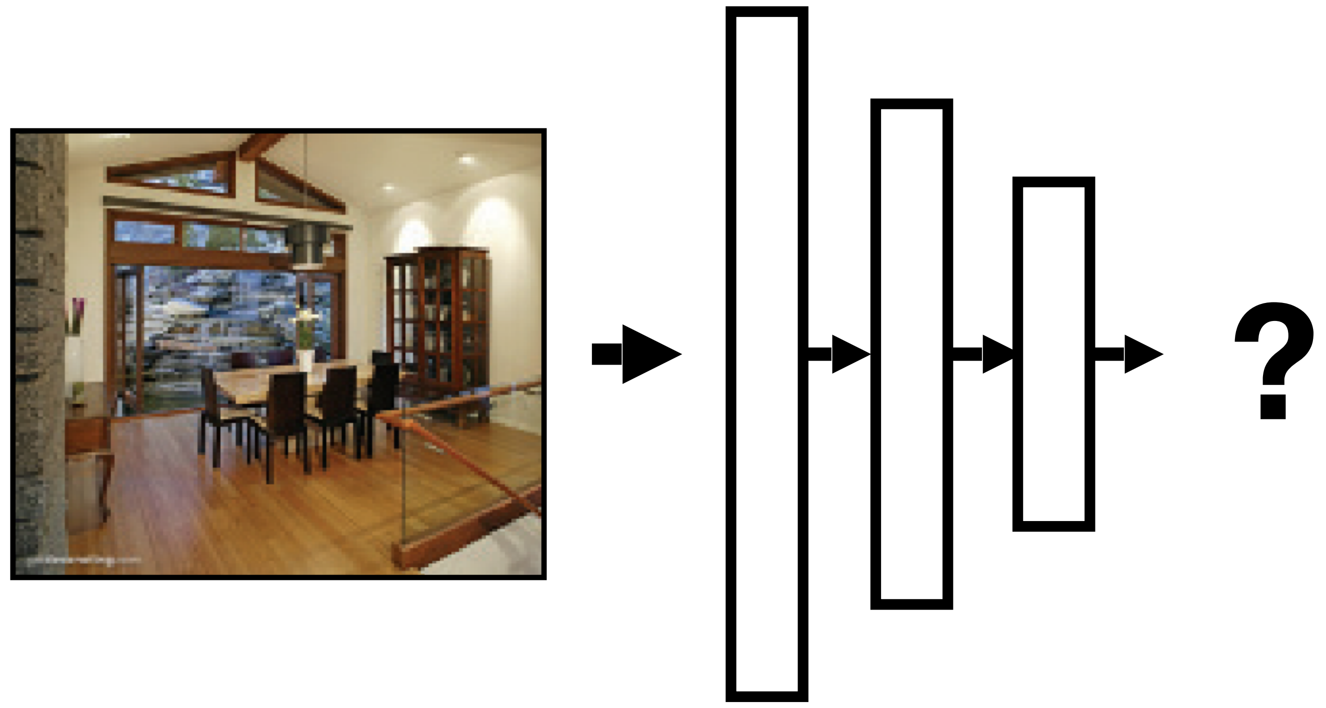
Place recognition



A little data

Testing

Place recognition



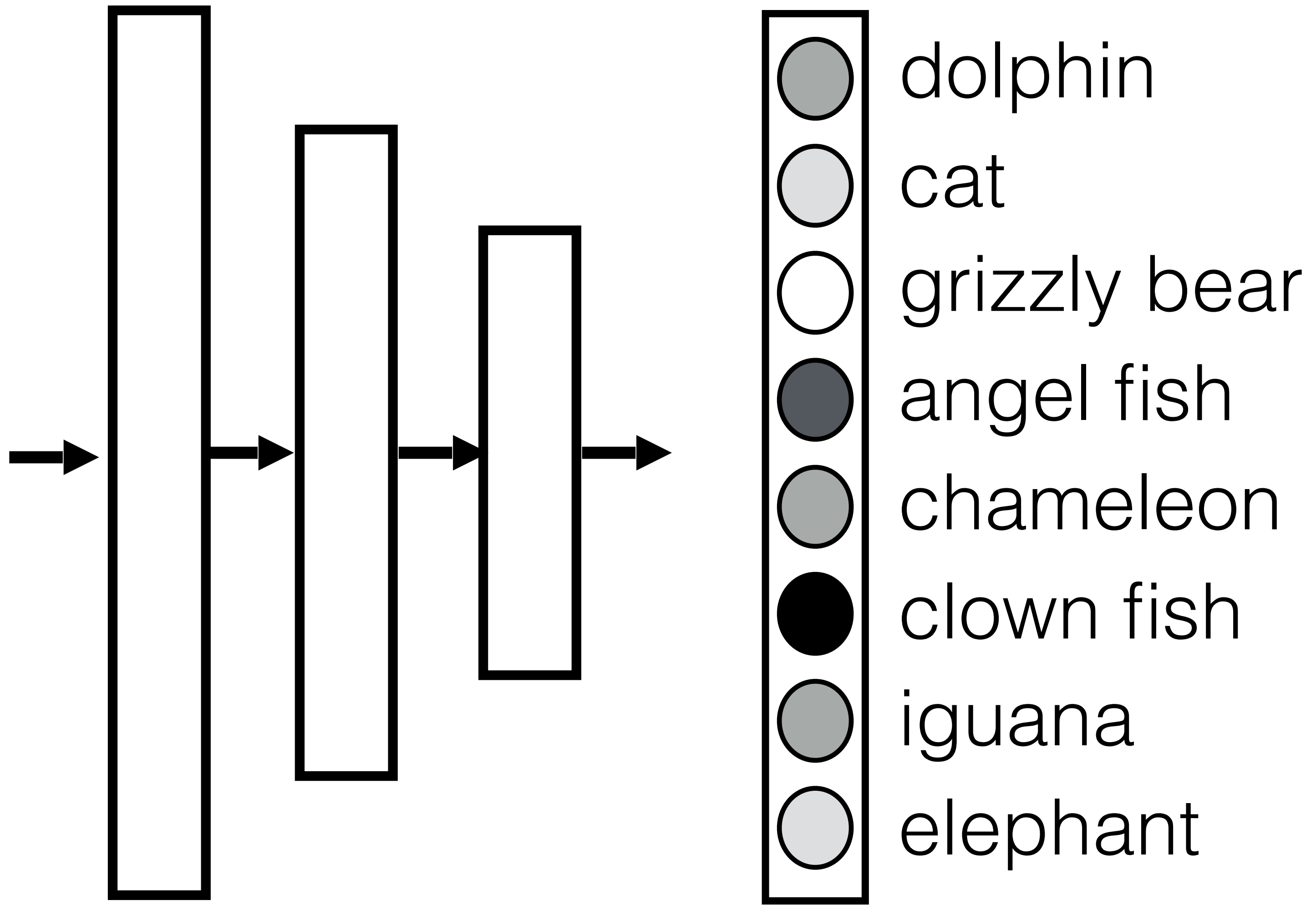
Finetuning starts with the representation learned on a previous task, and adapts it to perform well on a new task.

Source: Isola, Freeman, Torralba

Finetuning

Pretraining

Object recognition

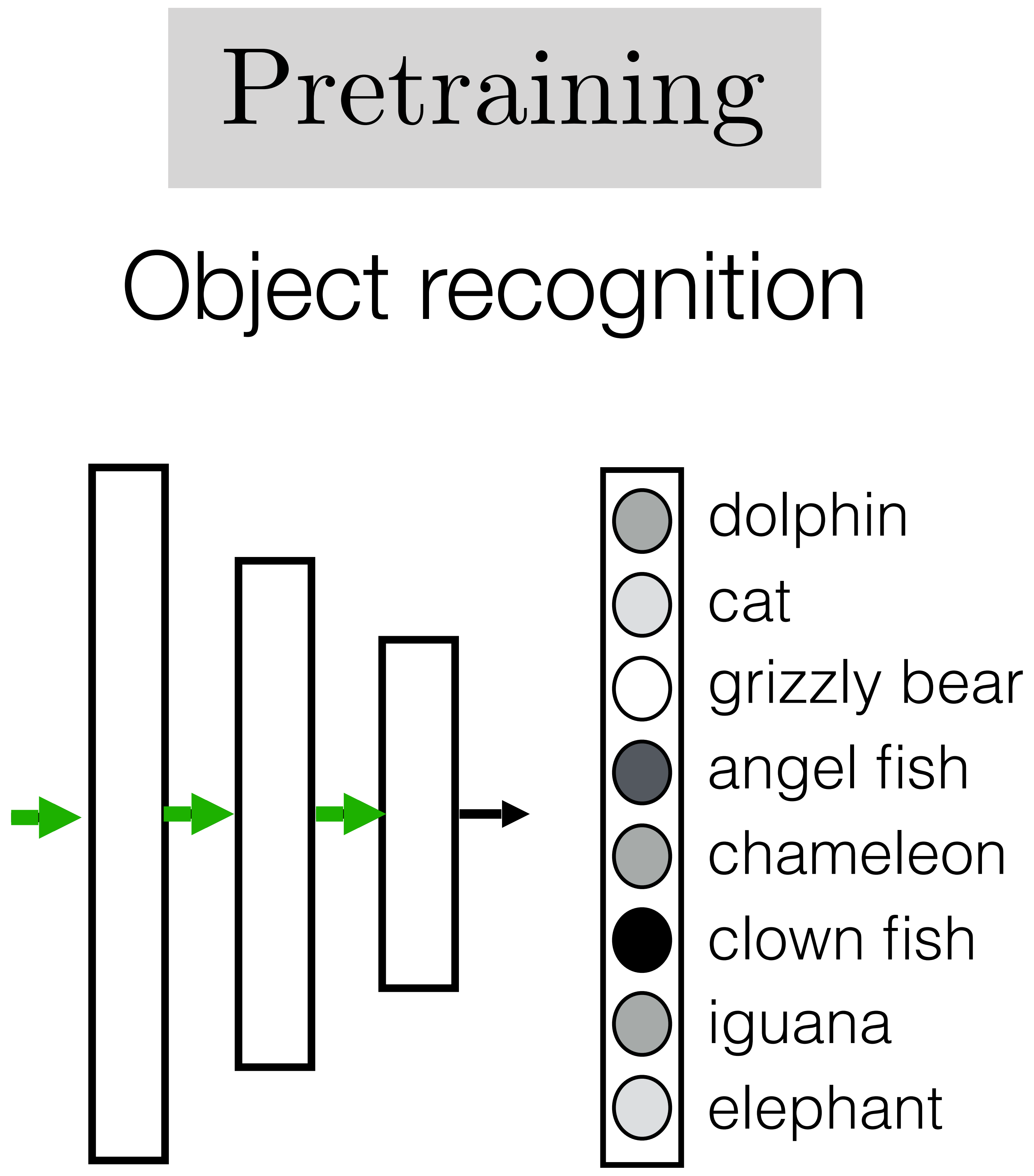


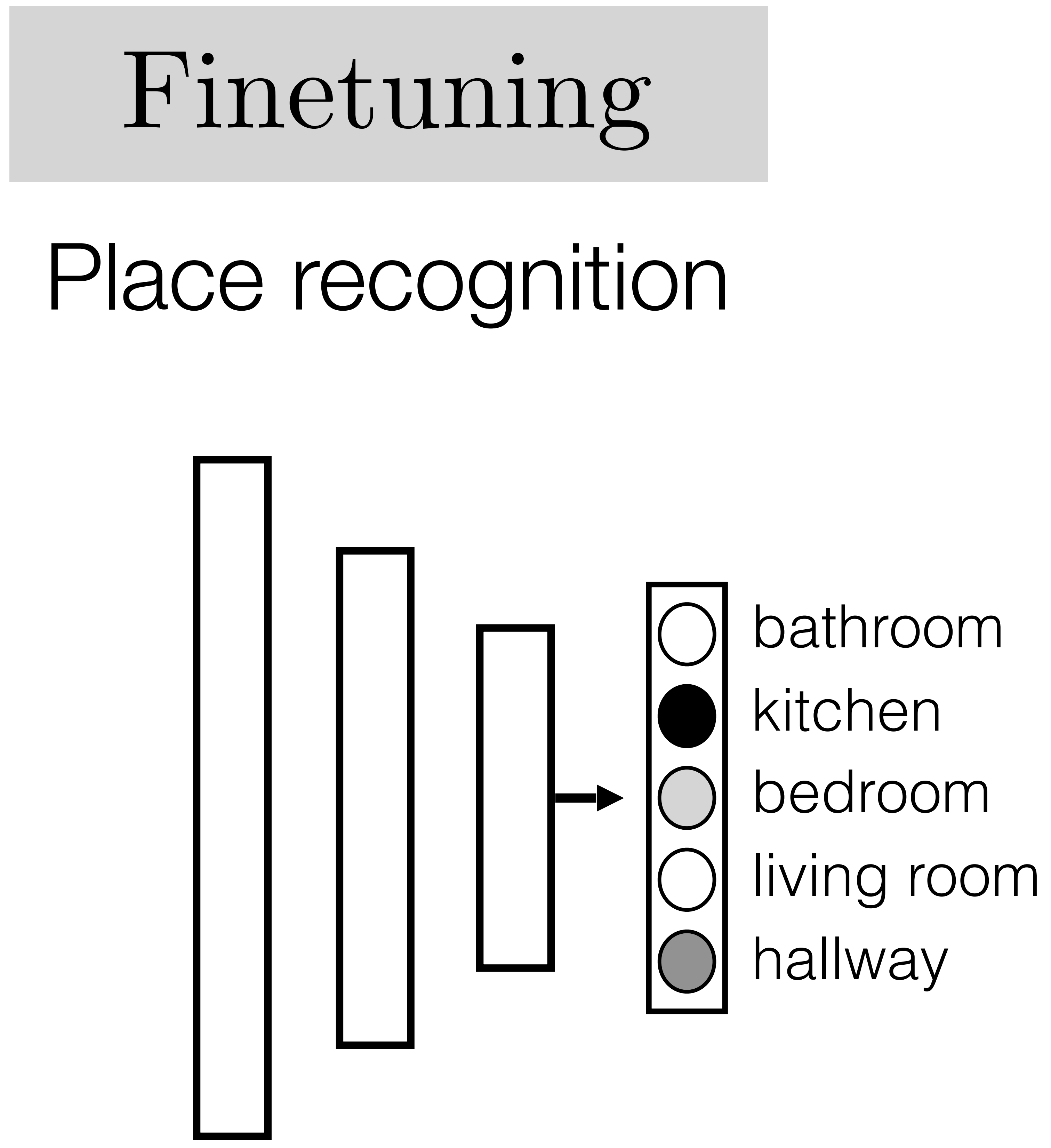
Finetuning

Place recognition

Source: Isola, Freeman, Torralba

Finetuning





The “learned representation” is just the weights and biases, so that’s what we transfer

Source: Isola, Freeman, Torralba

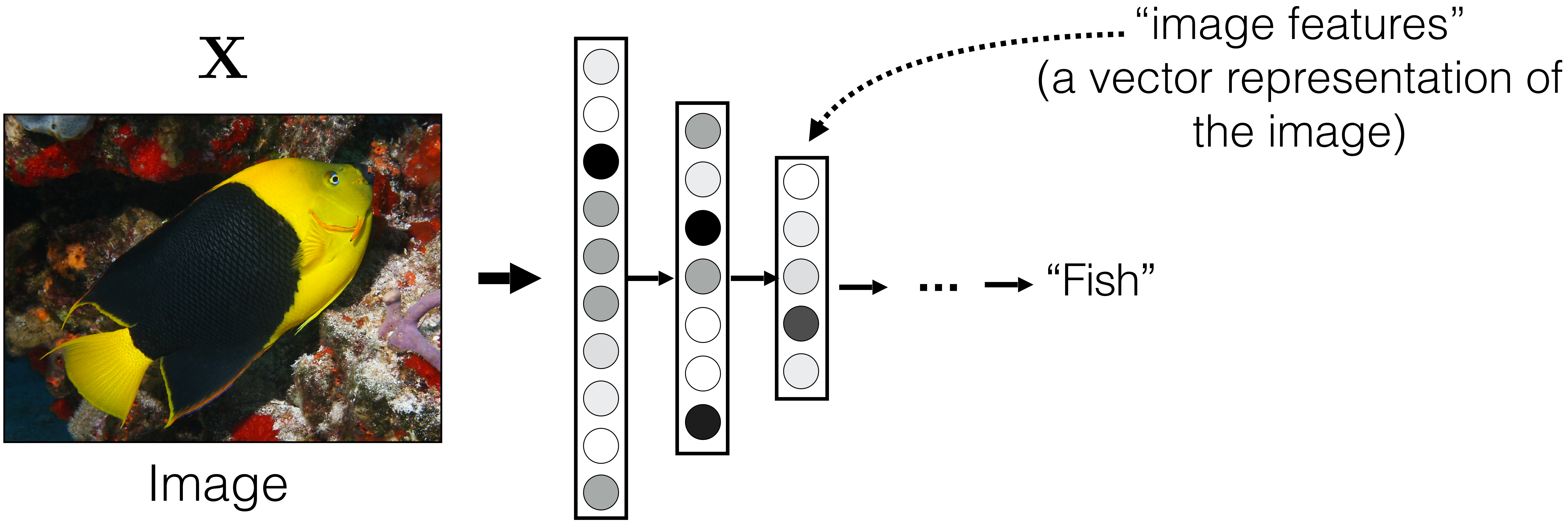
Finetuning

- Pretrain a network on task A (often object recognition), resulting in parameters **W**
- Initialize a second network with some or all of **W**
- Train the second network on task B, resulting in parameters **W’**
- Why would we expect this to work?

Source: Isola, Freeman, Torralba

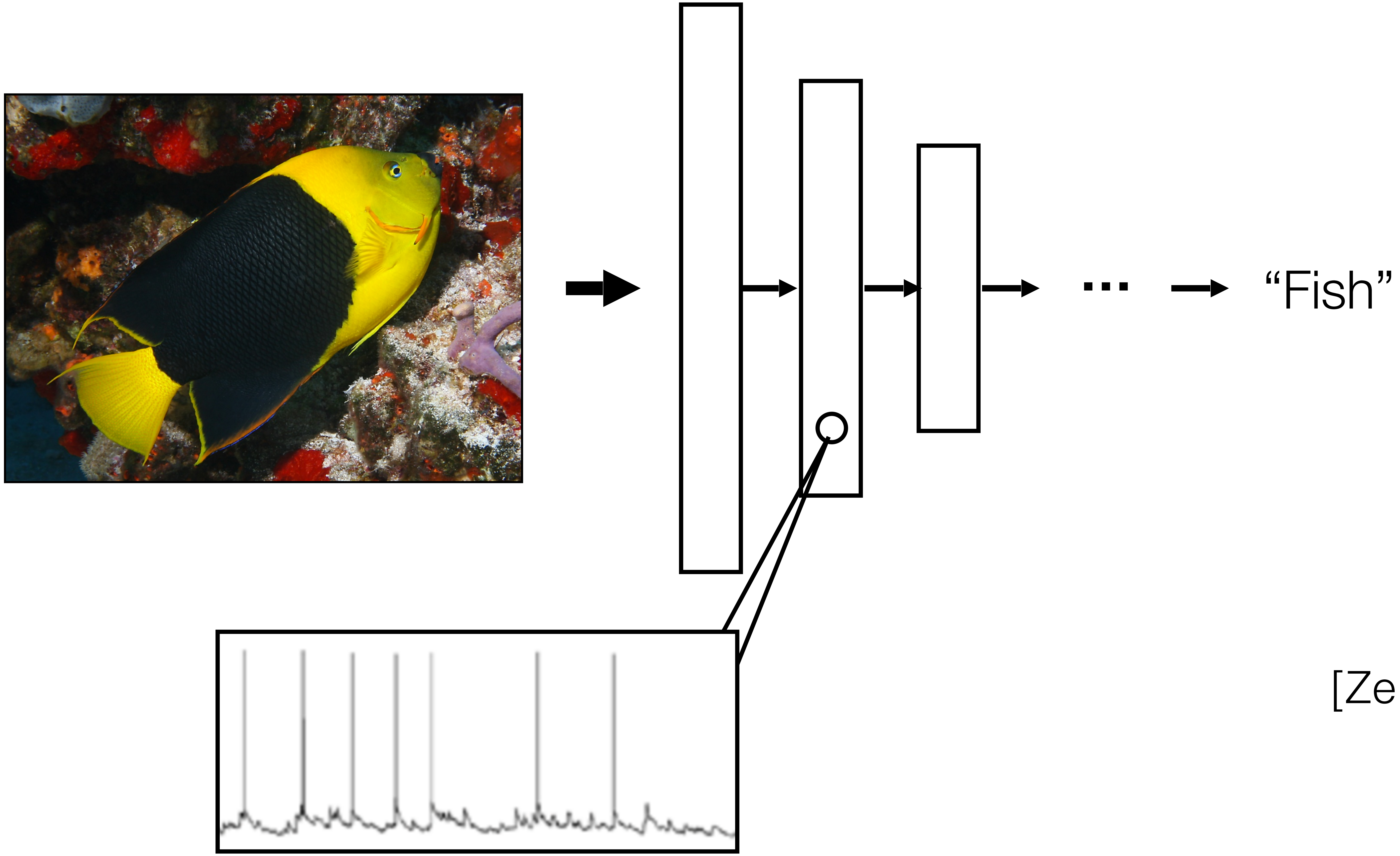
Visualizing representations

What do deep nets internally learn?



Source: Isola, Freeman, Torralba

Deep net “electrophysiology”



[Zeiler & Fergus, ECCV 2014]

[Zhou et al., ICLR 2015]

Visualizing and Understanding CNNs

[Zeiler and Fergus, 2014]

Gabor-like filters learned by **layer 1**

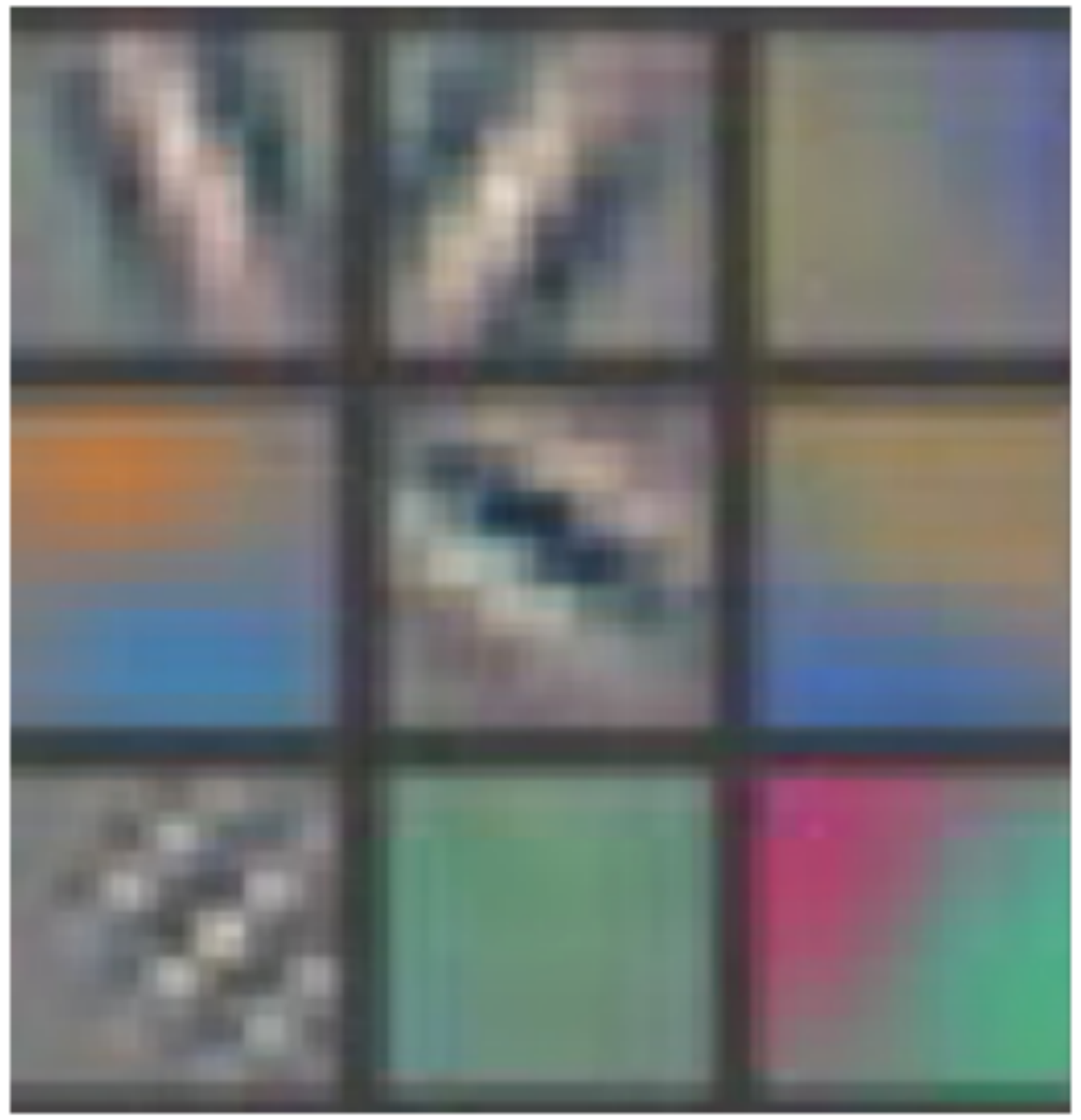
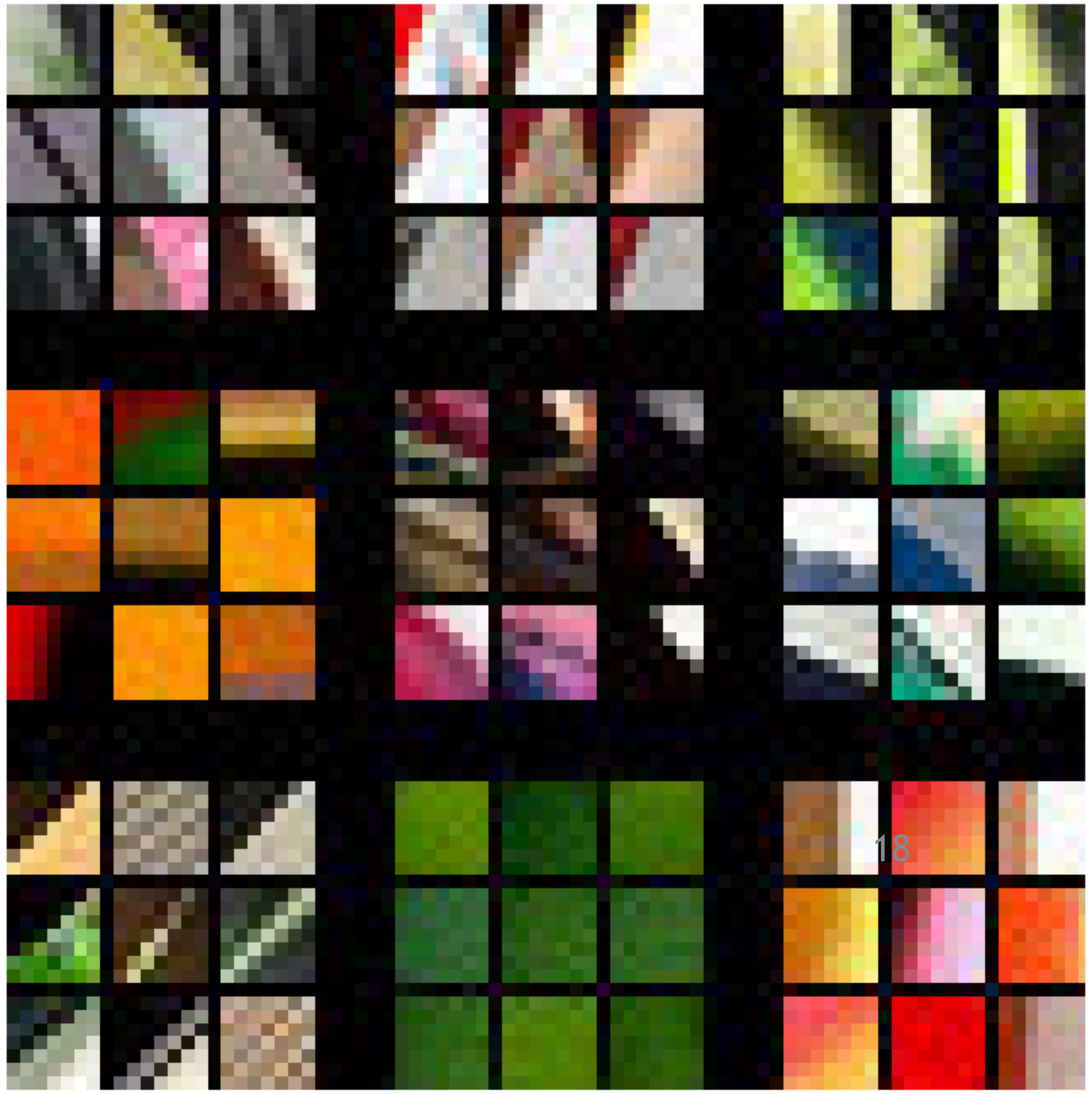


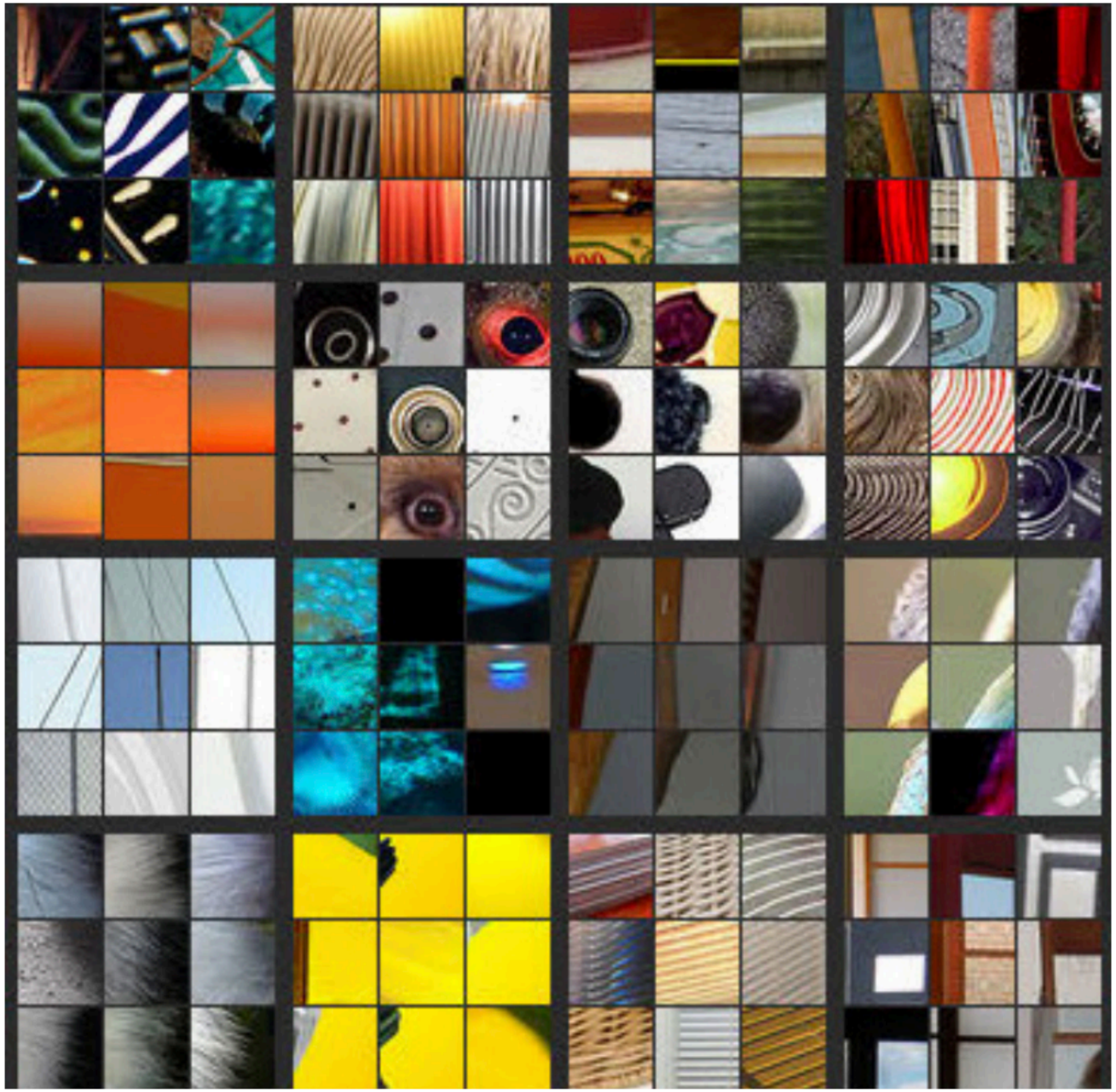
Image patches that activate each of the **layer 1** filters most strongly



Source: Isola, Freeman, Torralba

[Zeiler and Fergus, 2014]

Image patches that activate each of the **layer 2** neurons most strongly



Source: Isola, Freeman, Torralba

[Zeiler and Fergus, 2014]

Image patches that activate each of the **layer 3** neurons most strongly

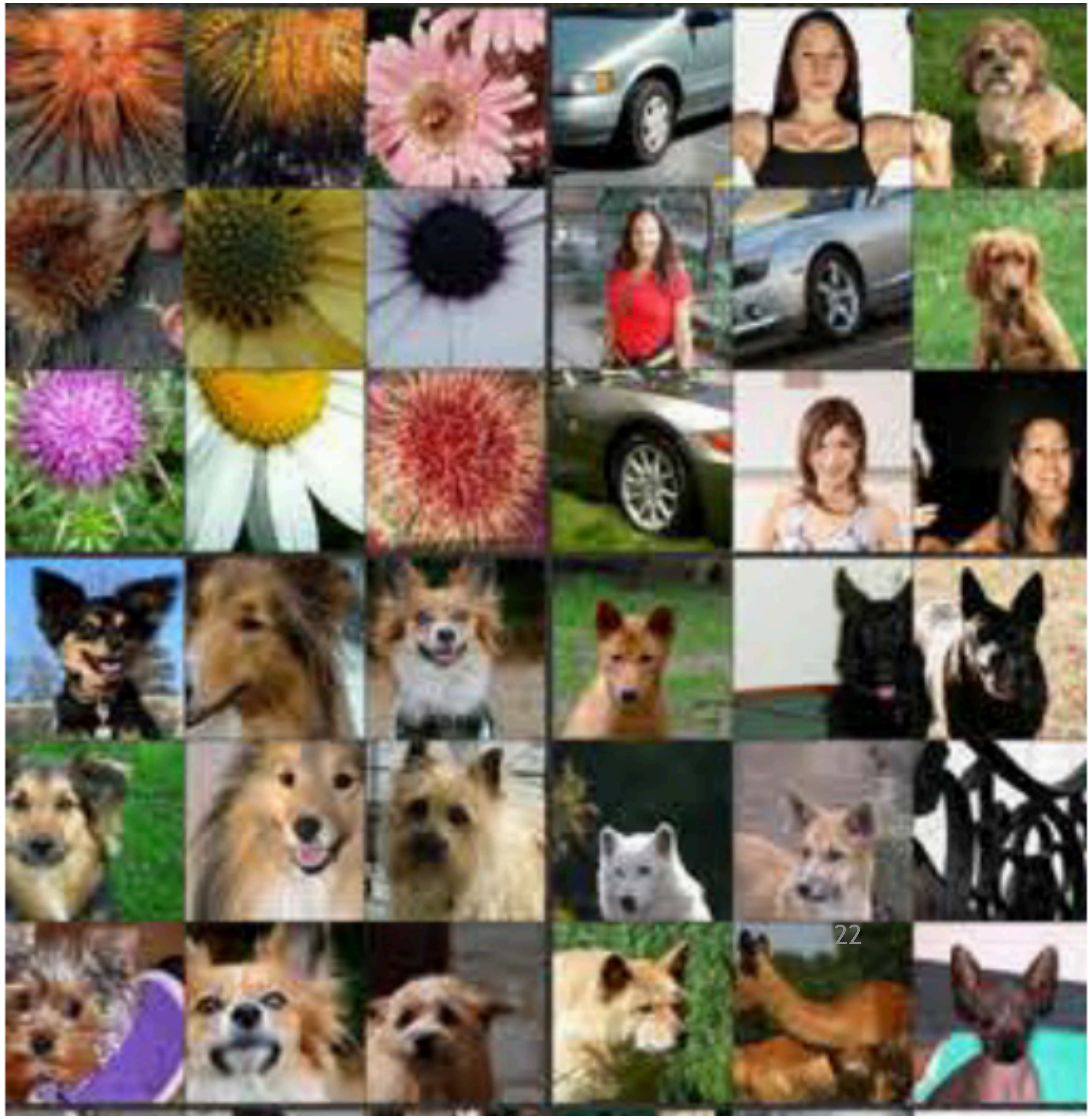
Source: Isola, Freeman, Torralba

[Zeiler and Fergus, 2014]

Image patches that activate each of the **layer 4** neurons most strongly

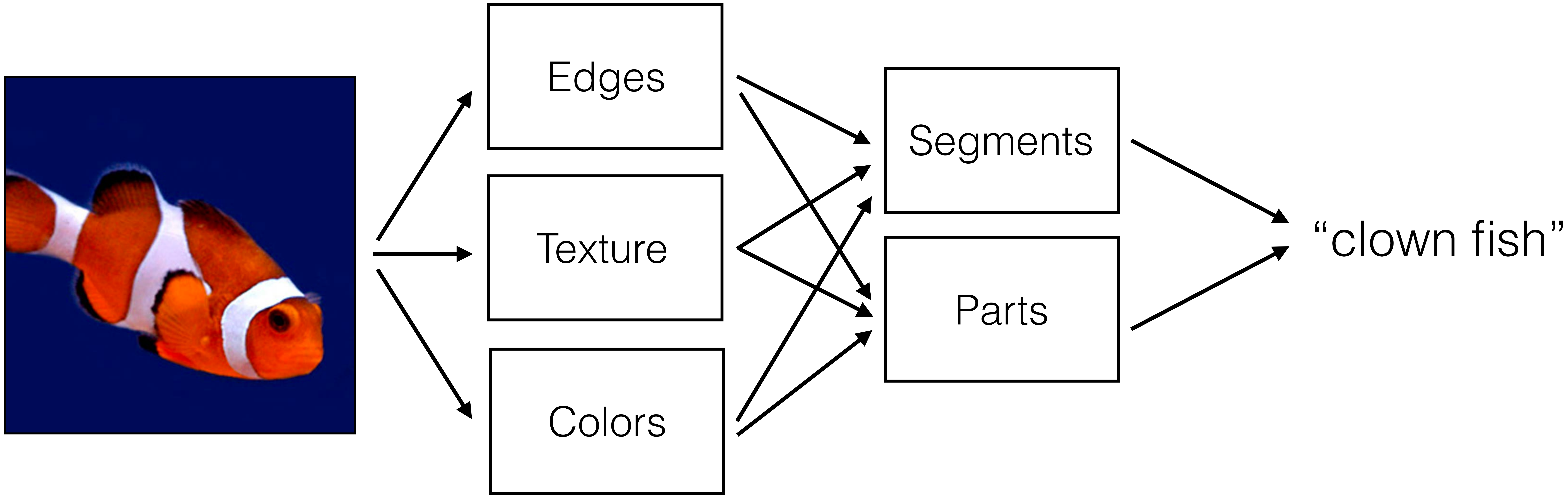
Source: Isola, Freeman, Torralba

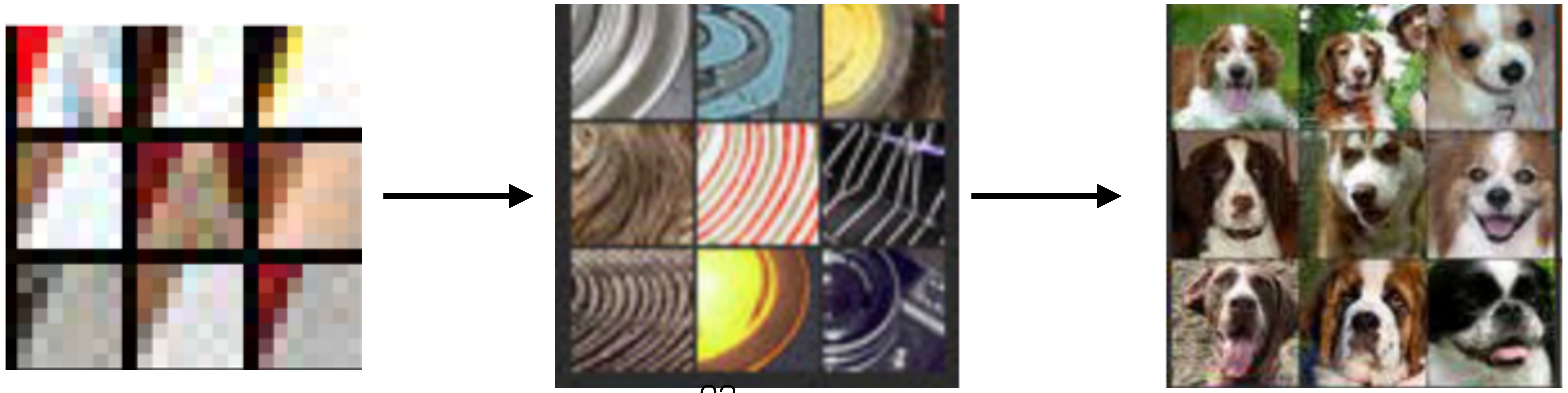
Image patches that activate each of the **layer 5** neurons most strongly



Source: Isola, Freeman, Torralba

CNNs learned the classical visual recognition pipeline



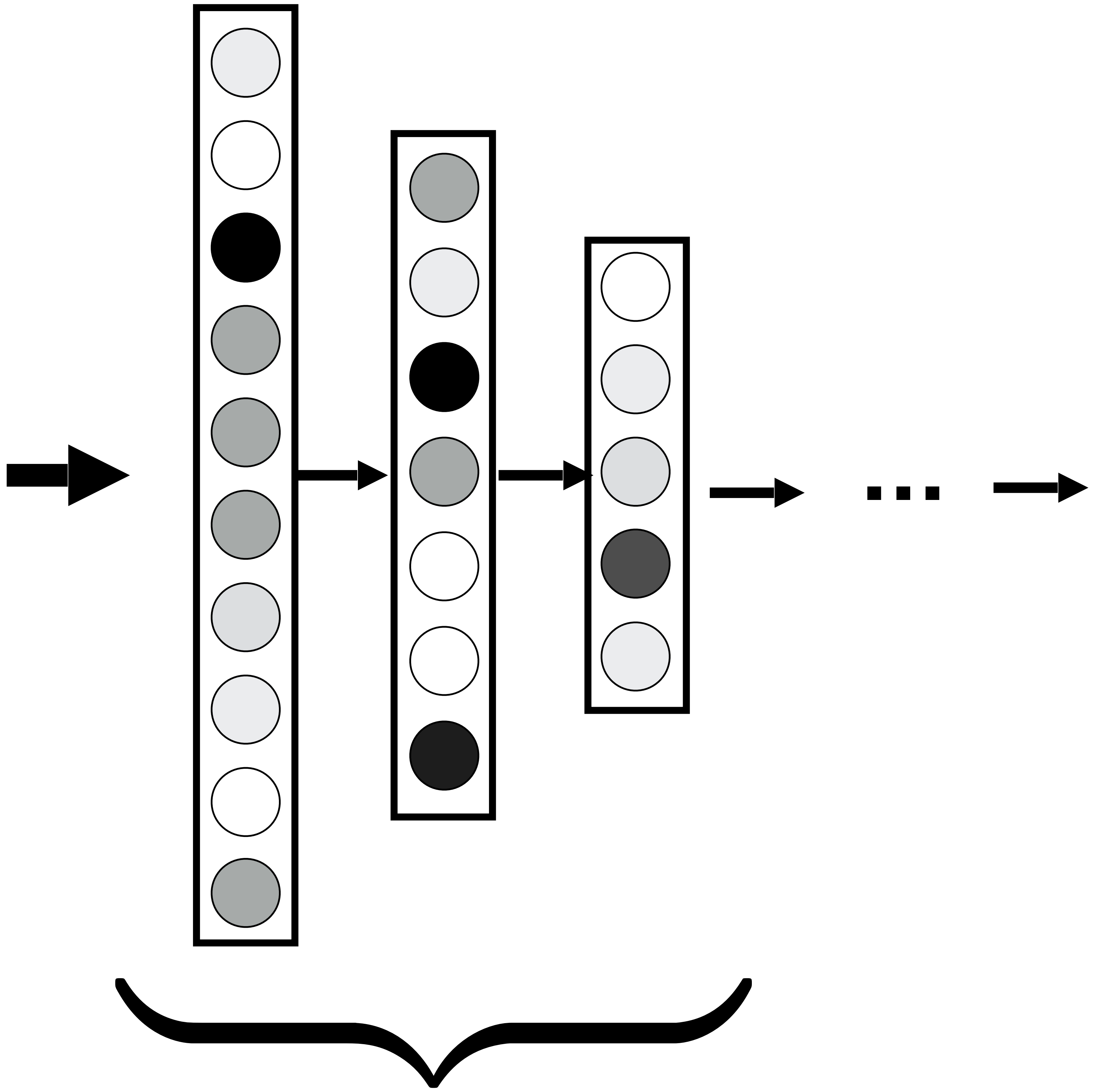


Source: Isola, Freeman, Torralba

What do deep nets internally learn?

$\mathbf{x}$

Image



Representations!

A CNN is a multiscale, hierarchical representation of data

Source: Isola, Freeman, Torralba

Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

AlexNet

- For each unit (neuron) in network, find which images it is most selective for (cause it to have highest activation)
- Find which pixels in these images are responsible by occluding regions and seeing which pixels, when occluded, cause activation to change the most
- Use a network trained on scene recognition

Source: Isola, Freeman, Torralba

Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

pool 1

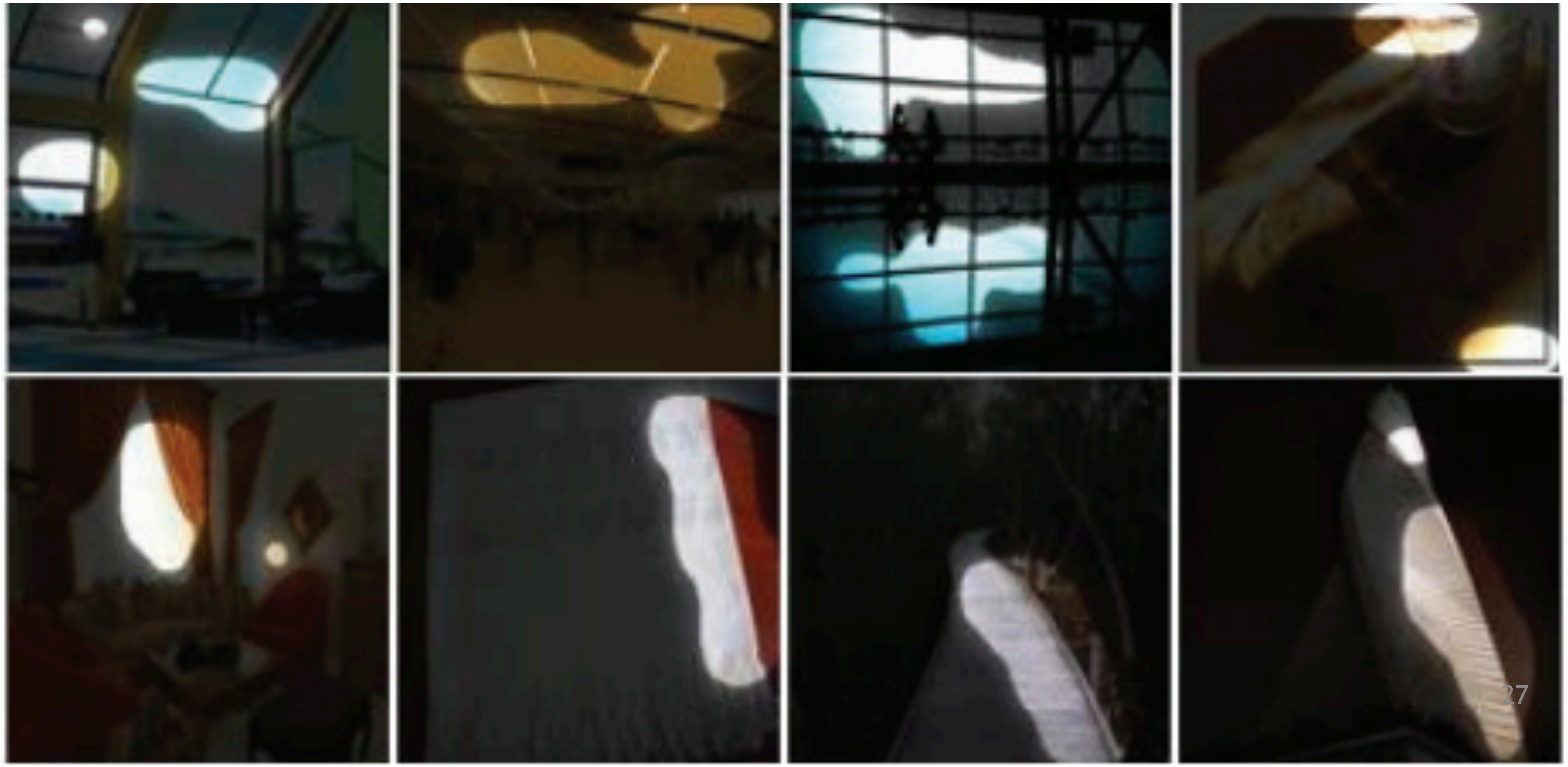
[http://people.csail.mit.edu/torralba/research/drawCNN/drawNet.html]

Source: Isola, Freeman, Torralba

Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

pool 2



Source: Isola, Freeman, Torralba

Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

conv 4

Source: Isola, Freeman, Torralba

Object Detectors Emergence in Deep Scene CNNs

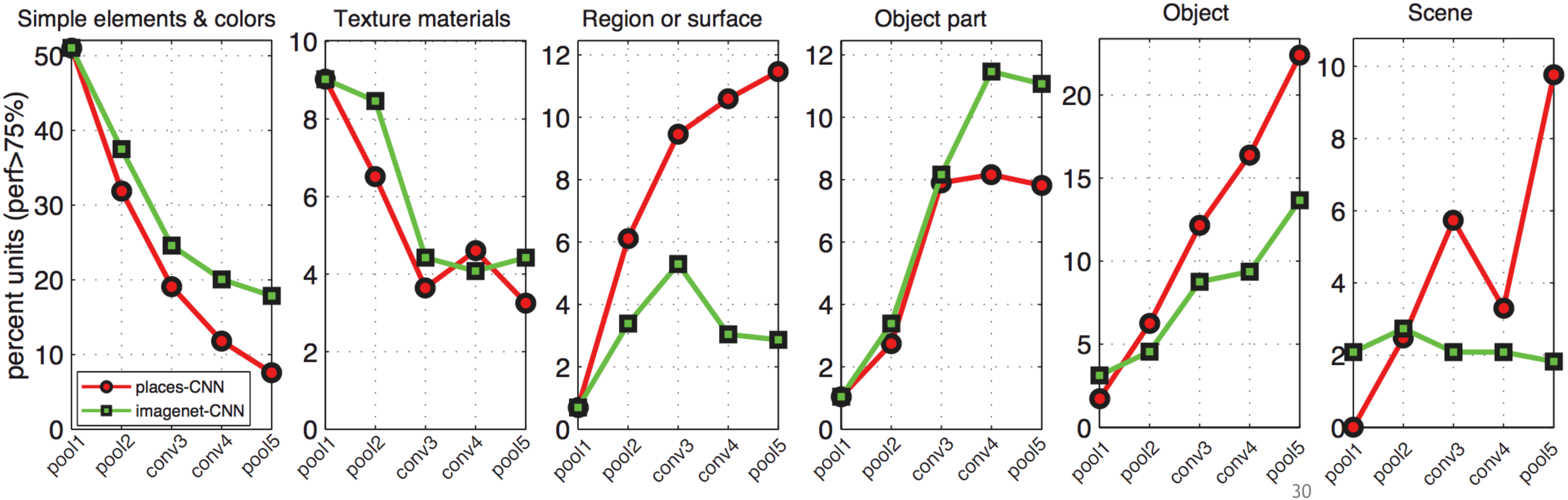
[Zhou et al., ICLR 2015]

pool 5

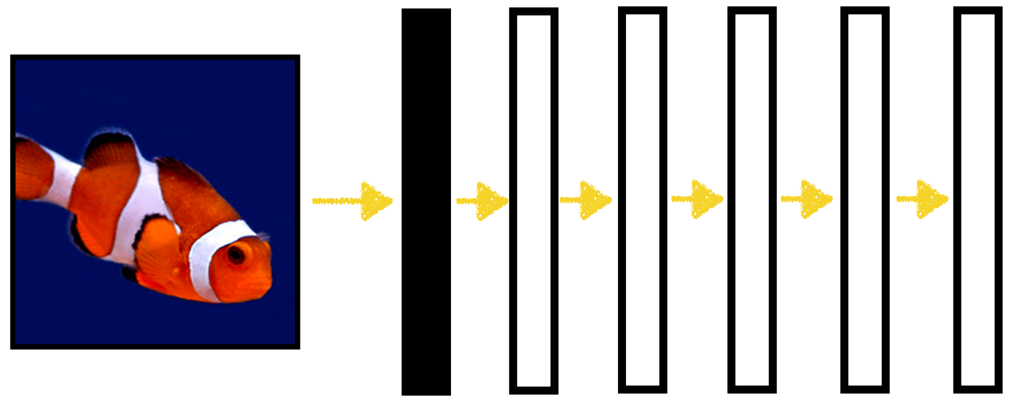
Source: Isola, Freeman, Torralba

Object Detectors Emergence in Deep Scene CNNs

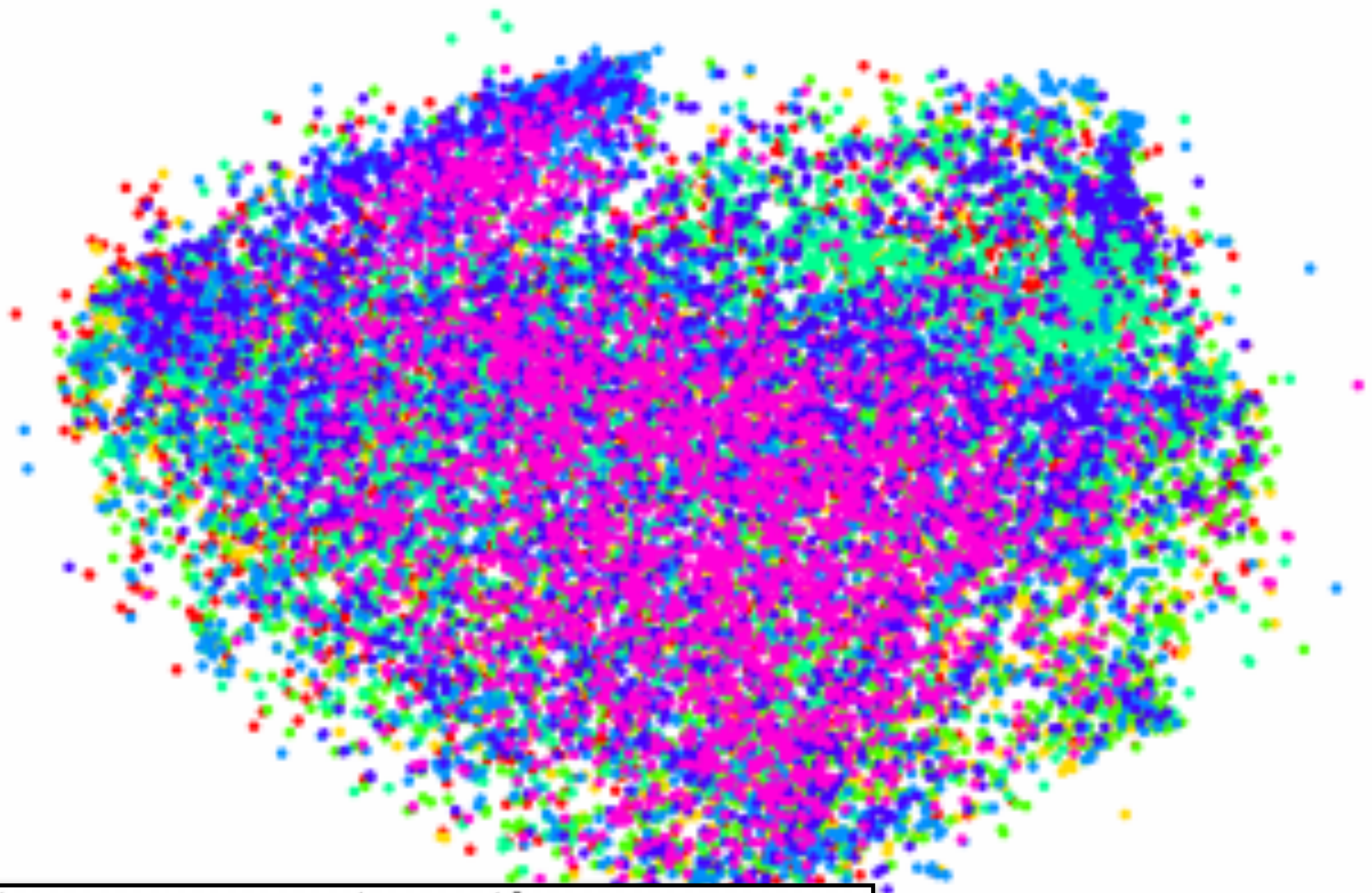
[Zhou et al., ICLR 2015]

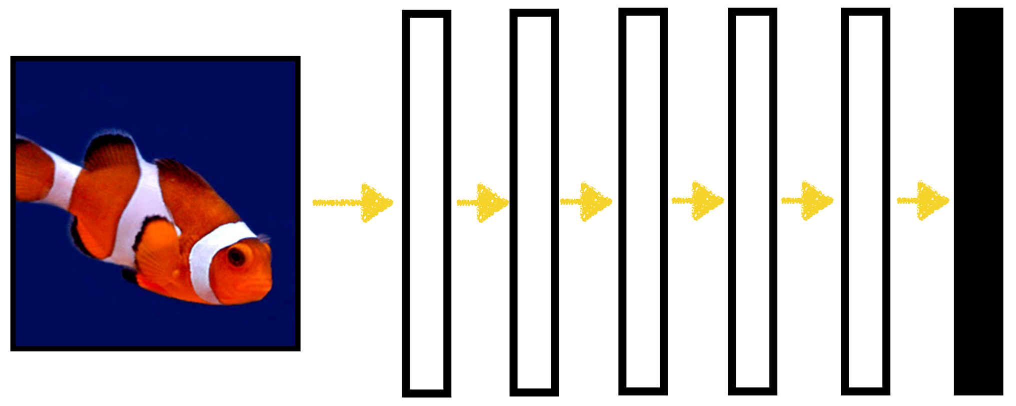


Source: Isola, Freeman, Torralba

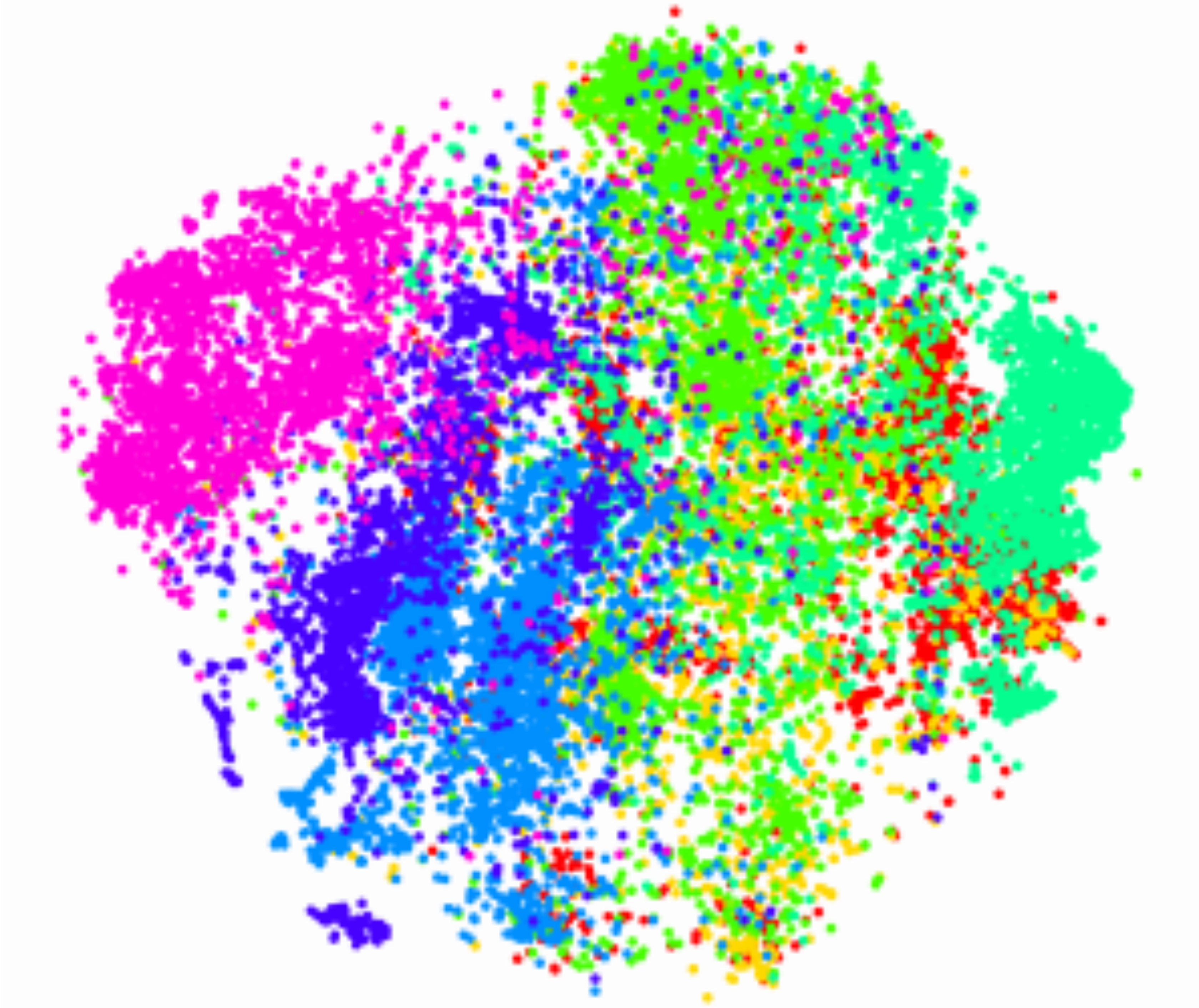


Layer 1 representation





Layer 6 representation



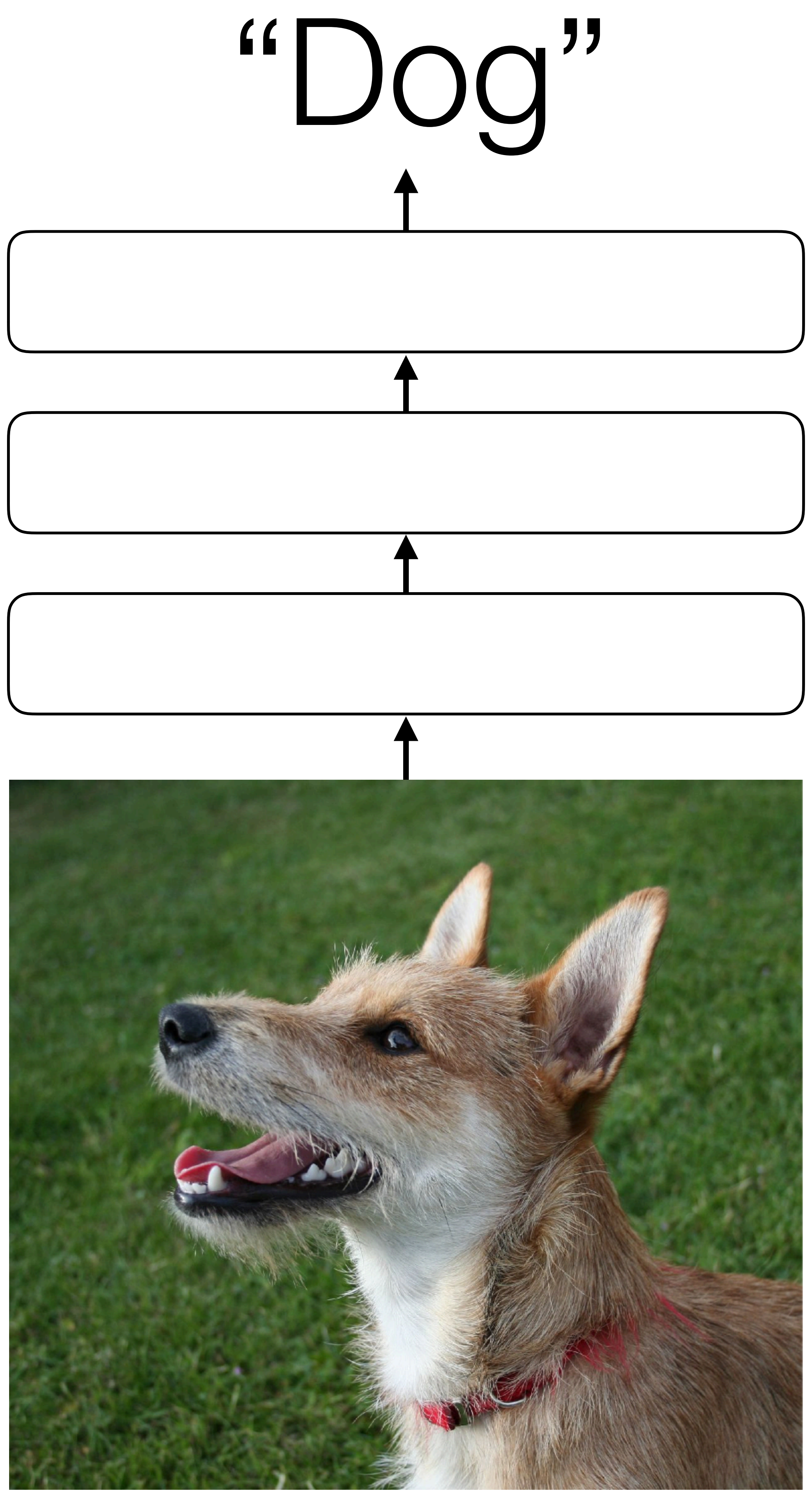
- structure, construction
- covering
- commodity, trade good, good
- conveyance, transport
- invertebrate
- bird
- hunting dog

[DeCAF, Donahue, Jia, et al. 2013]

[Visualization technique : t-sne, van der Maaten & Hinton, 2008]

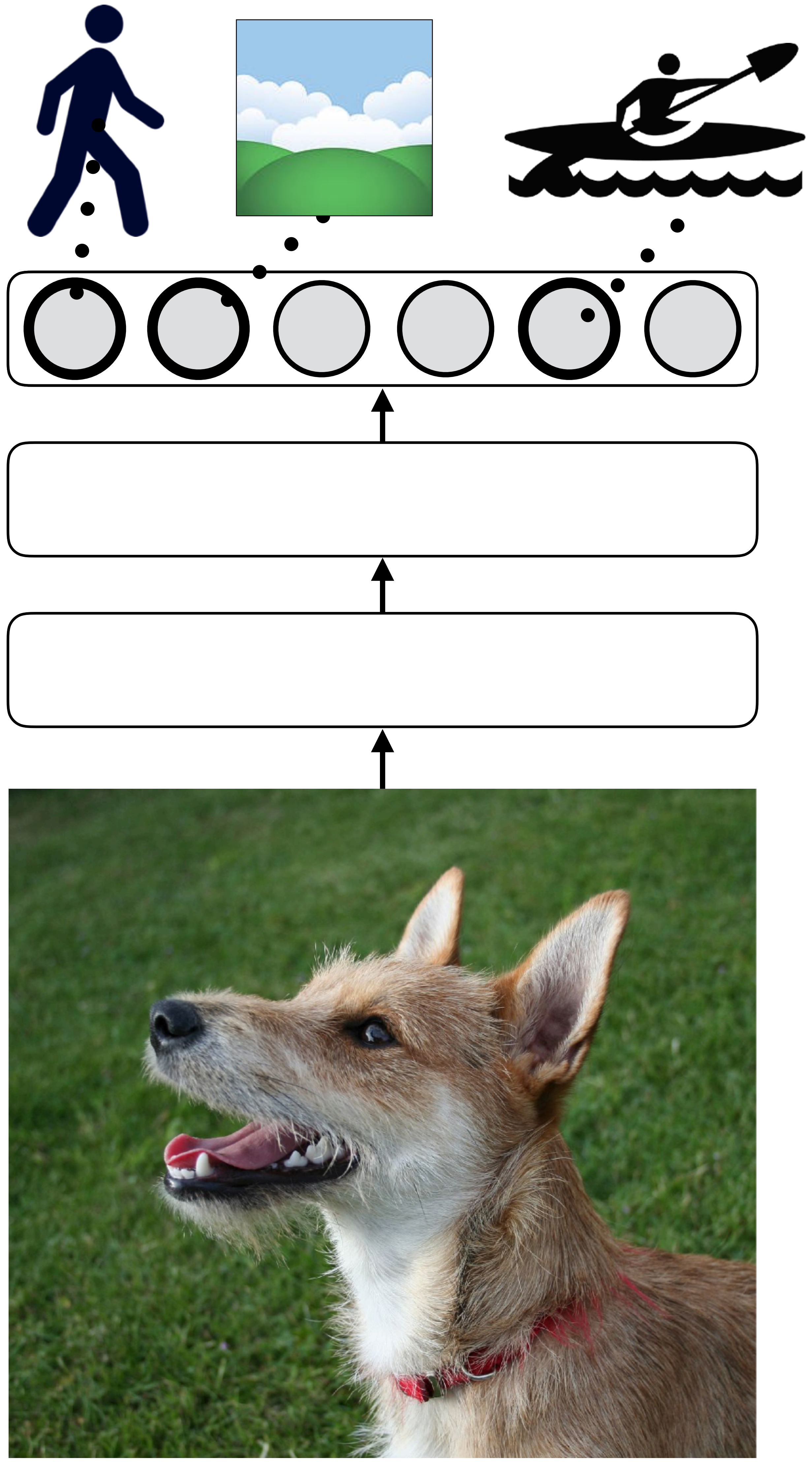
Source: Isola, Freeman, Torralba

Transferring CNN features



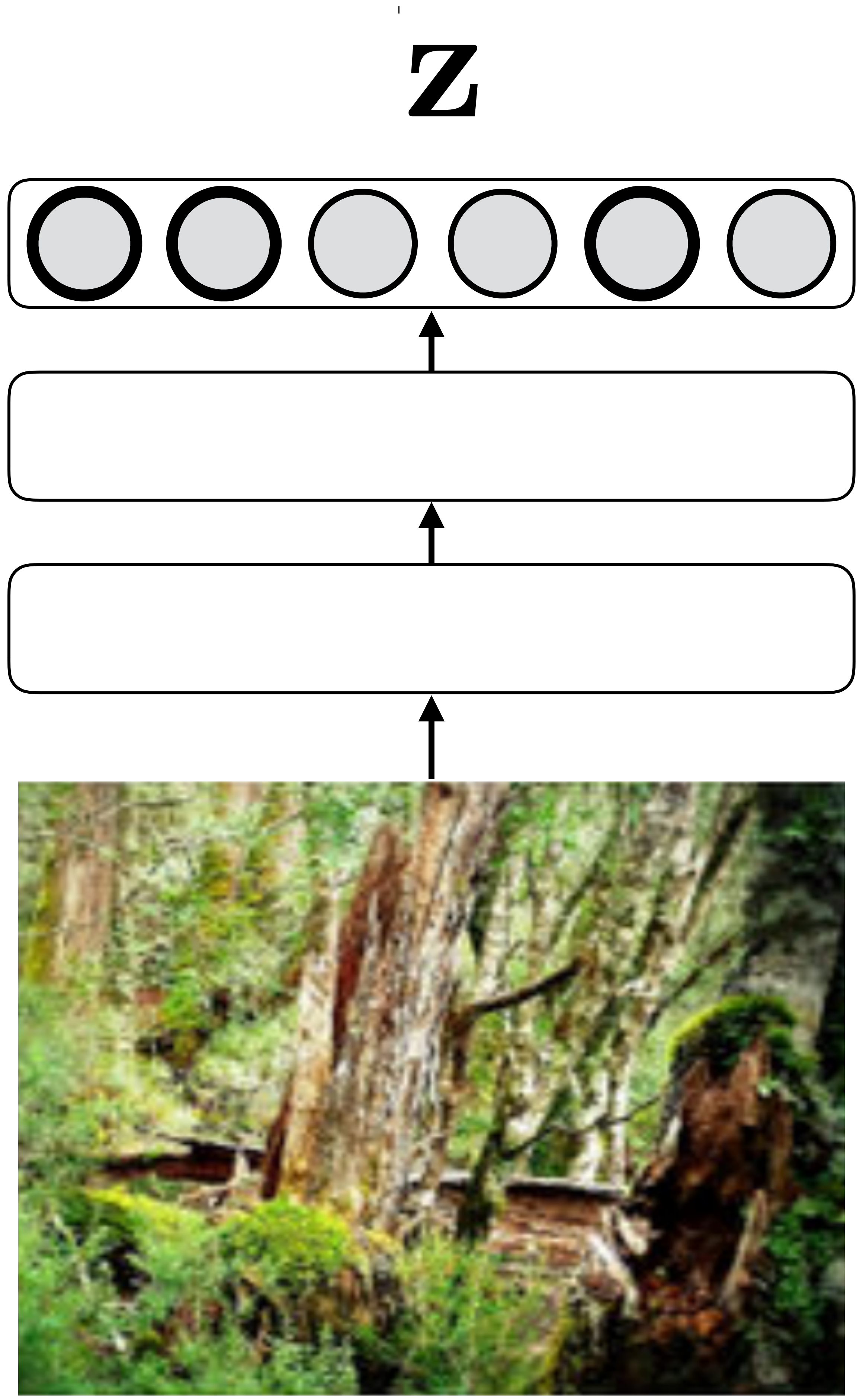
Object recognition net

Transferring CNN features



Object recognition net

Simple feature transfer



$(\mathbf{W}, \mathbf{b})$

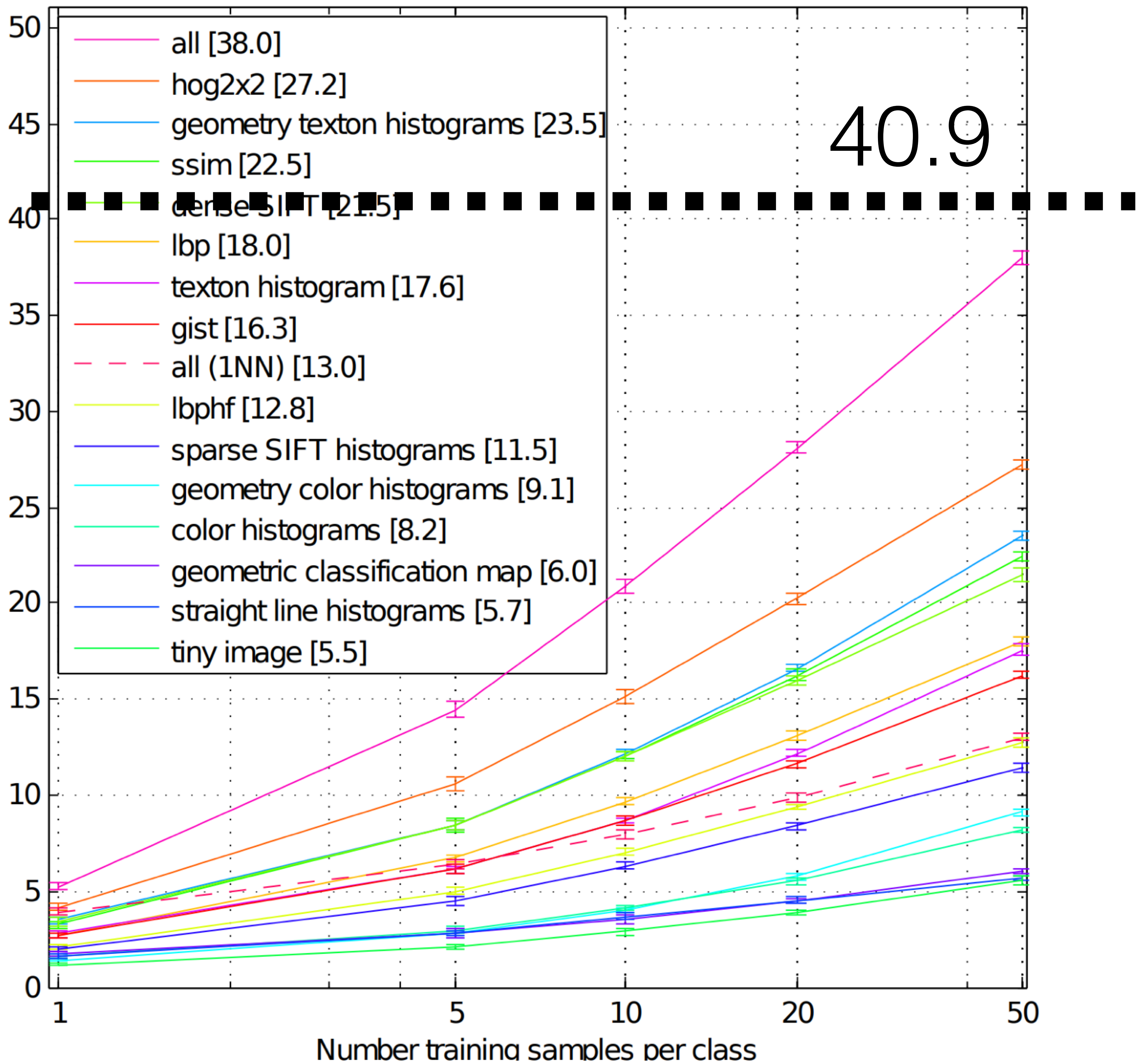
$\rightarrow$ Linear Classifier $\rightarrow$ y "Rainforest"

Logistic regression:

$$y = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b})$$

Transferring CNN features

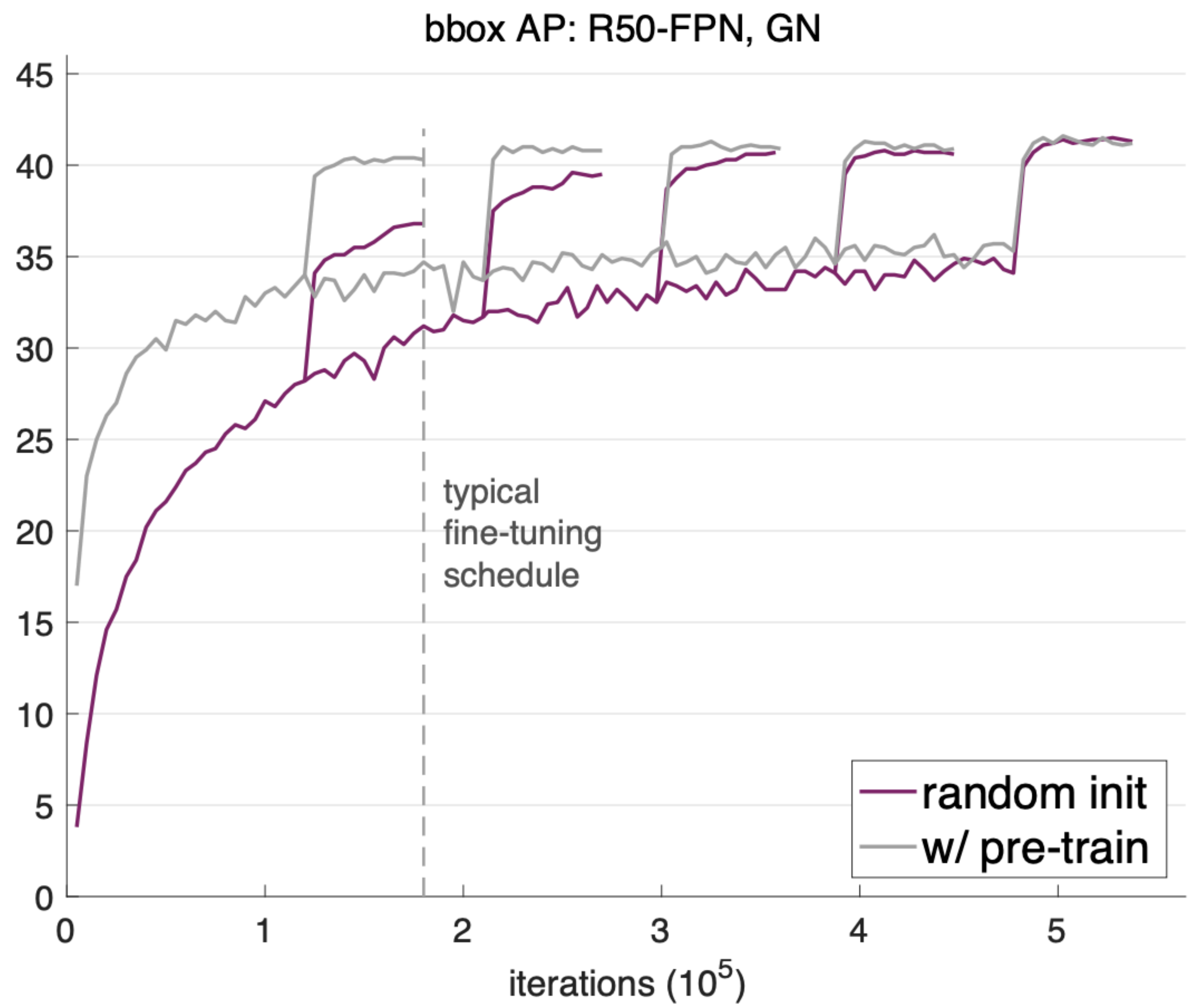
Hand-crafted features



[Xiao et al., CVPR 2010]

CNN features pretrained on ImageNet
+ linear classifier [Donahue et al. 2013]

Finetuning for object detection



- ImageNet pretraining speeds up object detection training.
- No change in accuracy for large datasets
- Big performance gains for small/medium datasets though (e.g. 1K examples per class)!

[He et al. 2018]

How do we learn good representations?

Supervised object recognition

image X → **Learner** → "Fish"

label Y

Source: Isola, Freeman, Torralba

Supervised object recognition

Learner

“Fish”

image X

label Y

Source: Isola, Freeman, Torralba

Supervised object recognition

Learner

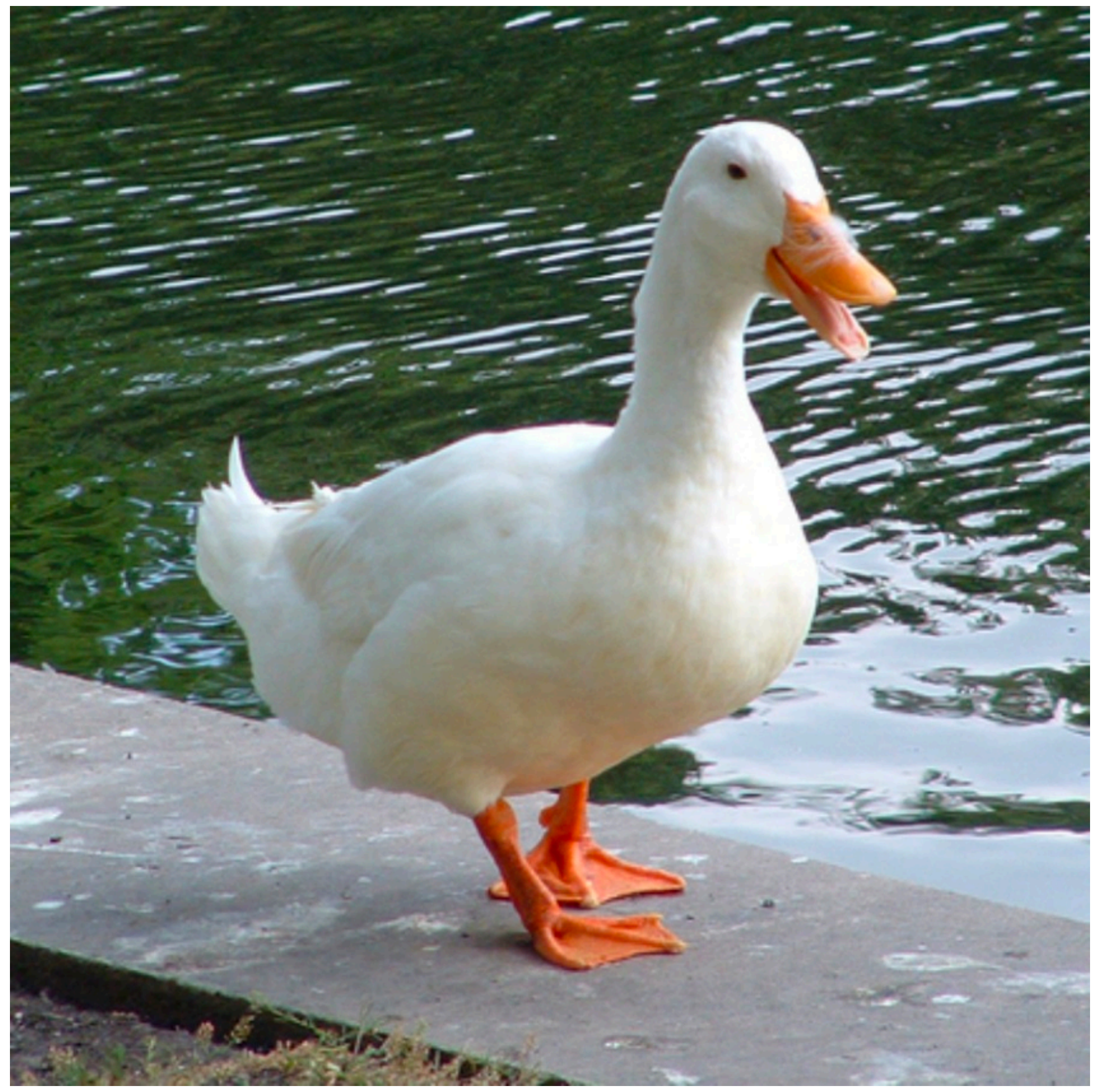
“Fish”

image X

label Y

Source: Isola, Freeman, Torralba

Supervised object recognition



Learner

"Duck"

⋮

image X

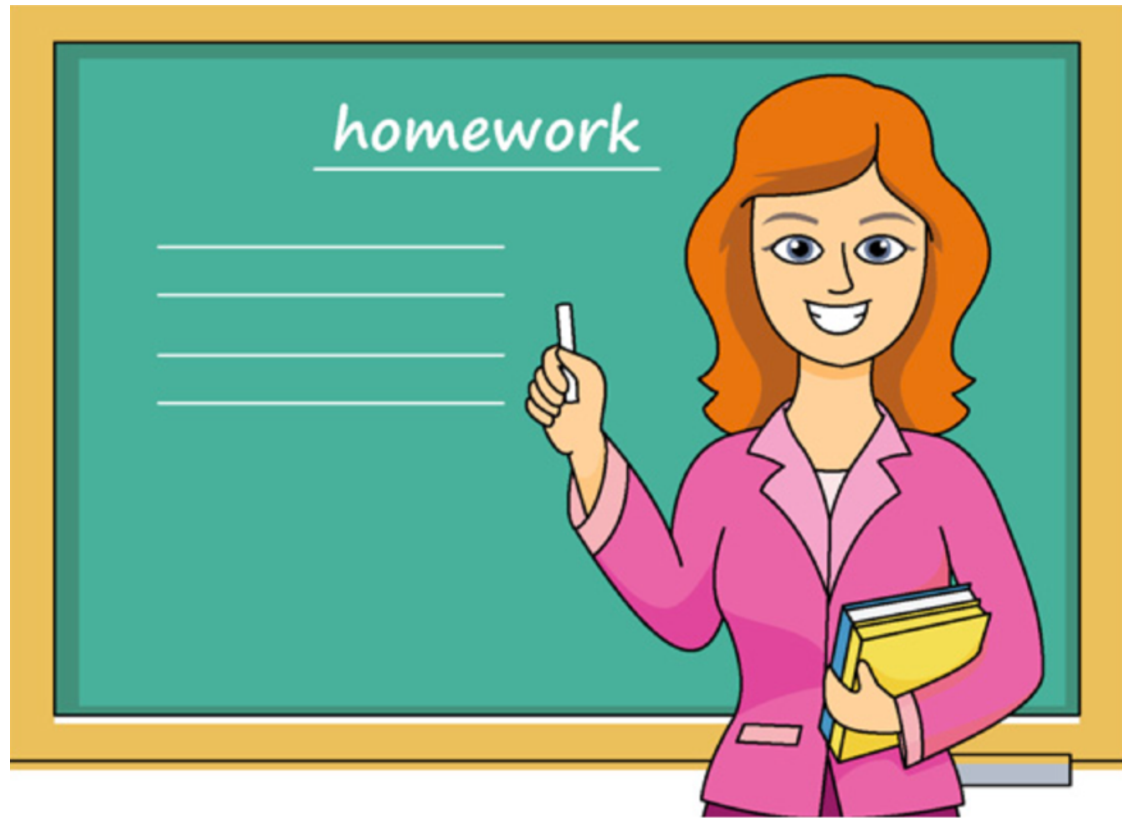
label Y

Source: Isola, Freeman, Torralba

Supervised computer vision

Hand-curated training data

+ Informative
- Expensive
- Limited to teacher's knowledge



Vision in nature

Raw unlabeled training data

+ Cheap
- Noisy
- Harder to interpret

Source: Isola, Freeman, Torralba

Learning from examples

(aka **supervised learning**)

Training data

$$\{x_1, y_1\}$$
$$\{x_2, y_2\} \quad \rightarrow \quad \boxed{\text{Learner}} \quad \rightarrow \quad f : X \rightarrow Y$$
$$\{x_3, y_3\}$$
$$\ldots$$

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i)$$

Source: Isola, Freeman, Torralba

Representation Learning

Data

$\{x_1\}$

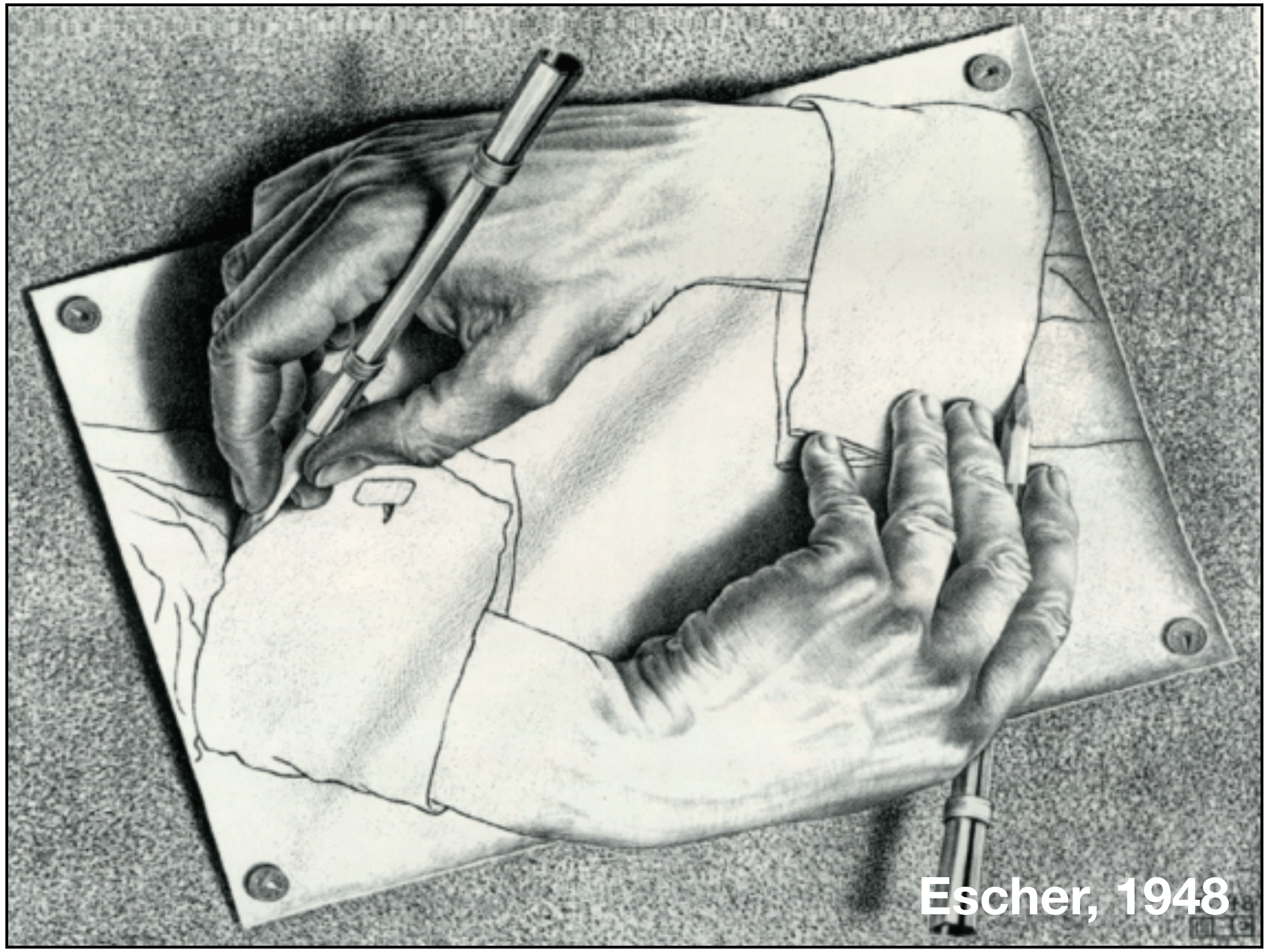
$\{x_2\}$ $\rightarrow$ Learner $\rightarrow$ Representations

$\{x_3\}$

$\dots$

Source: Isola, Freeman, Torralba

Self-supervised learning



Common trick:

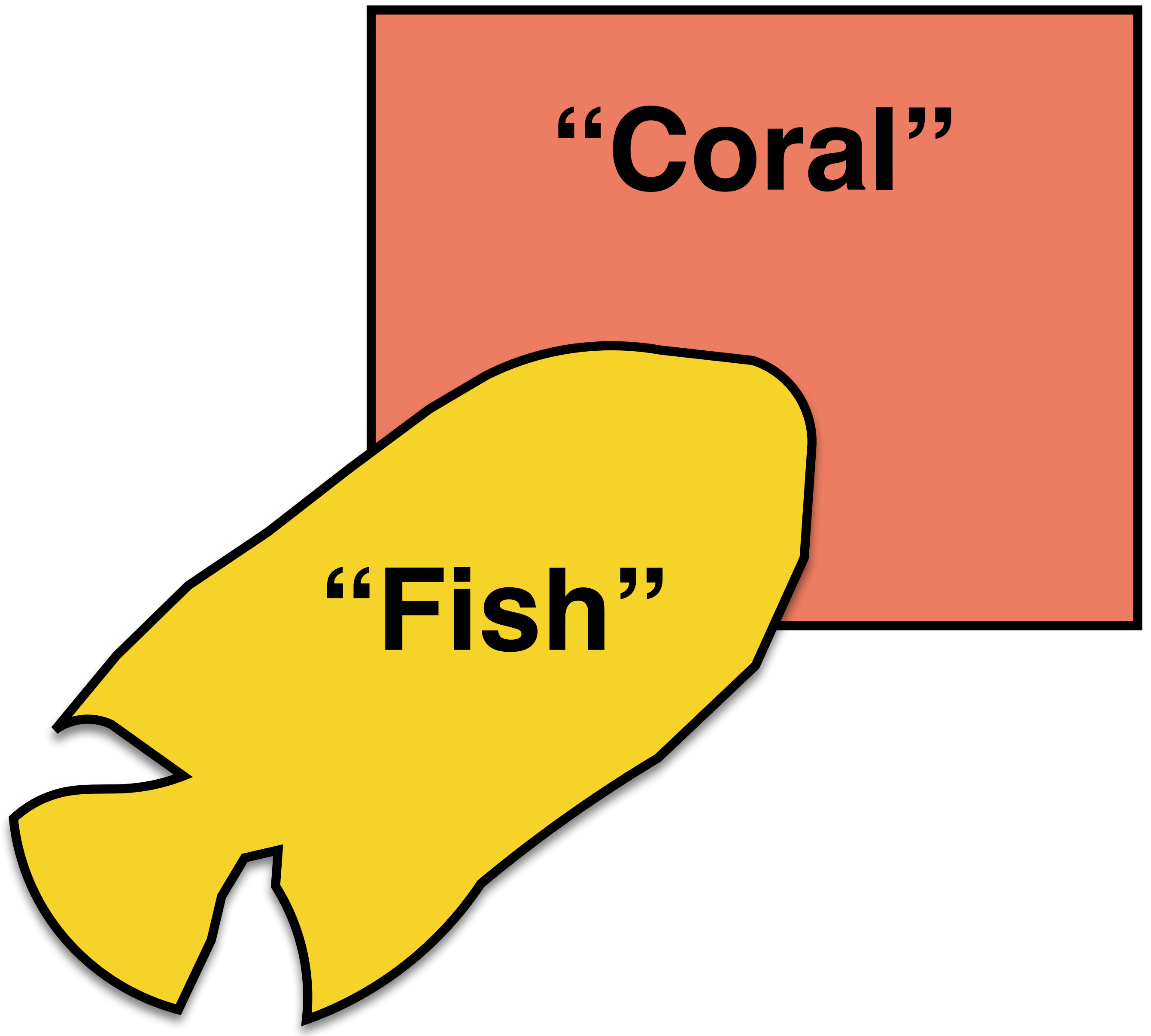
- Convert “unsupervised” problem into “supervised” empirical risk minimization
- Do so by cooking up “labels” (prediction targets) from the raw data itself
- Designing new algorithms still takes a lot of trial and error.

Source: Isola, Freeman, Torralba

Unsupervised Representation Learning

$\mathbf{X}$

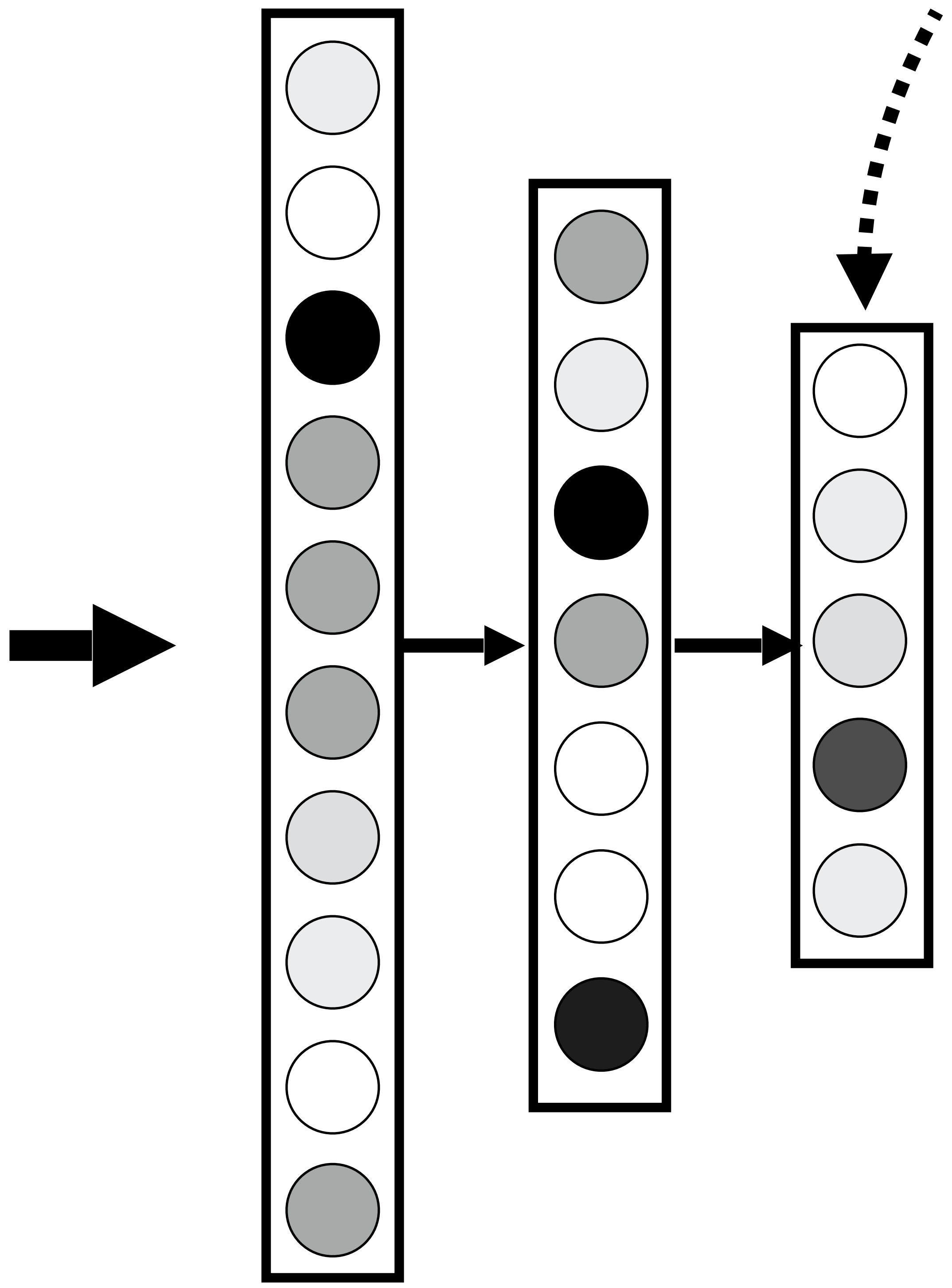
Image



Compact mental representation

Source: Isola, Freeman, Torralba

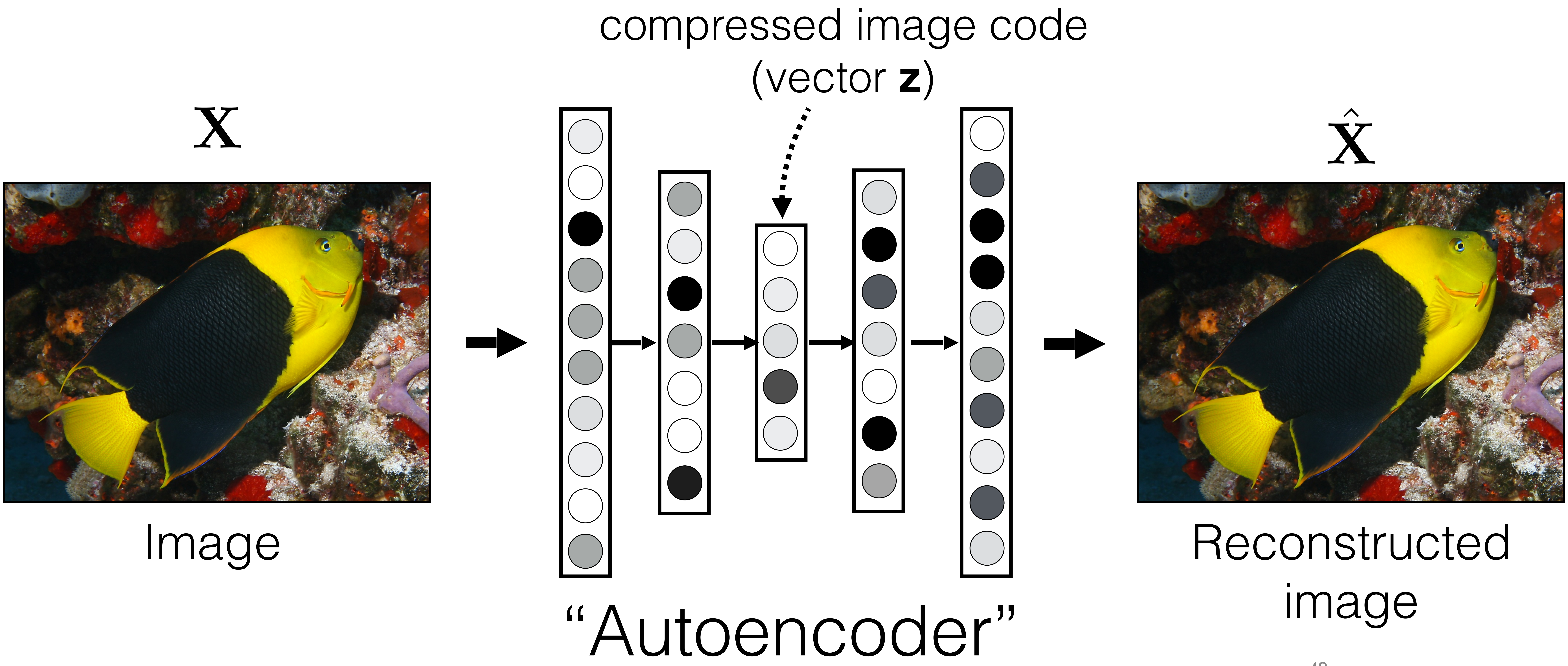
Unsupervised Representation Learning



Image

Source: Isola, Freeman, Torralba

Unsupervised Representation Learning



[e.g., Hinton & Salakhutdinov, Science 2006]

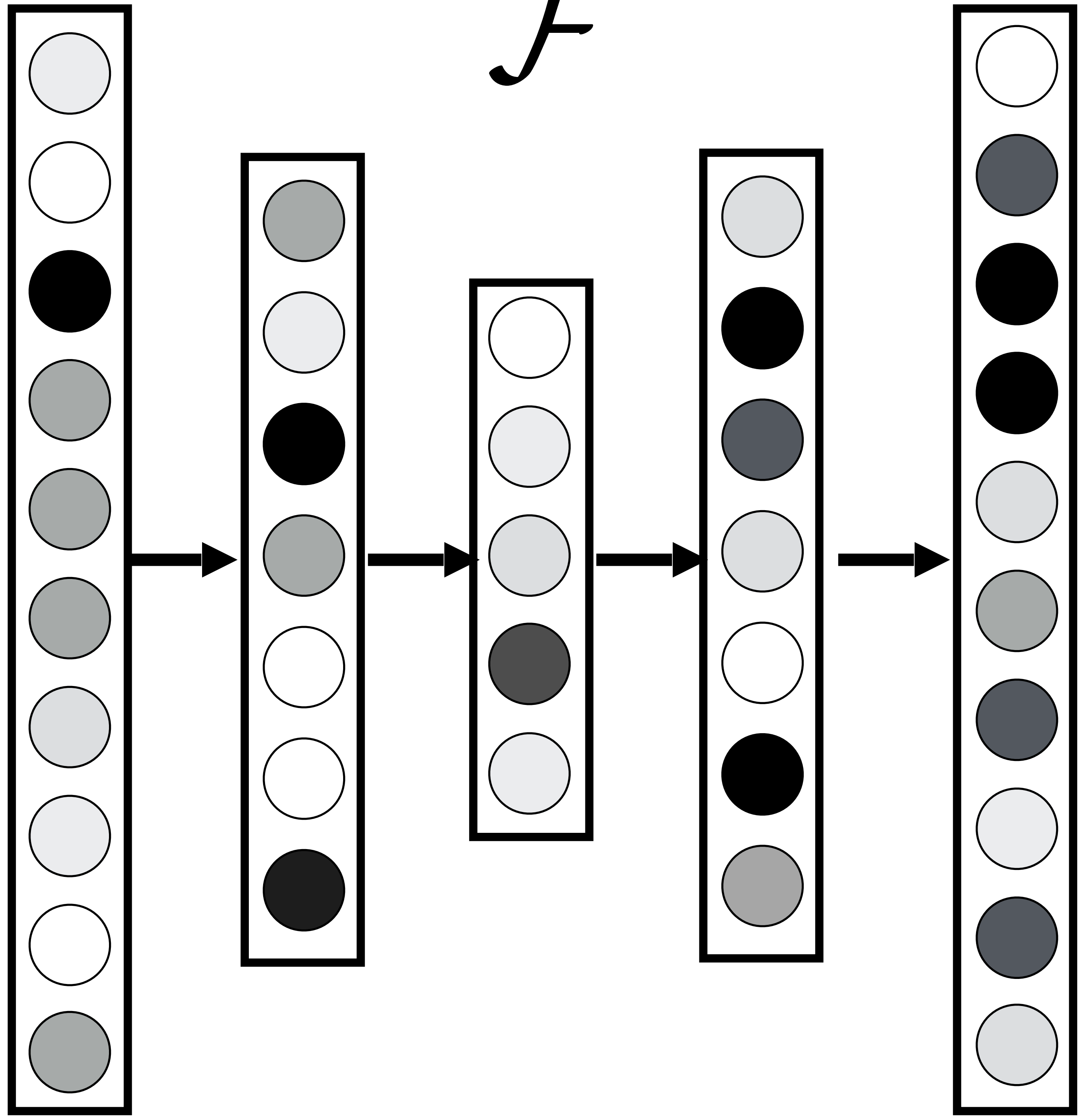
Source: Isola, Freeman, Torralba

Autoencoder

$\mathbf{X}$

Image

$\mathcal{F}$

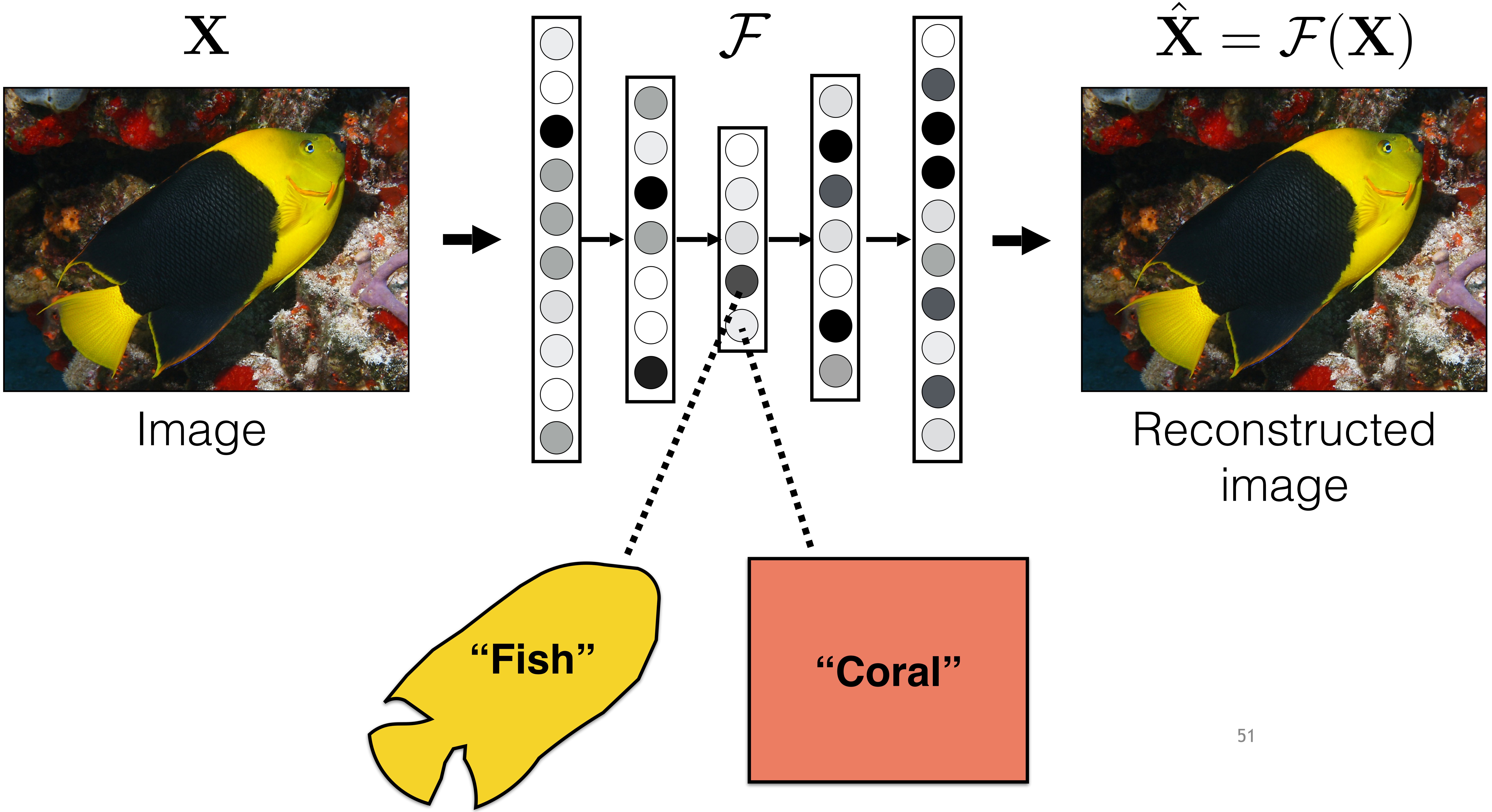


$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{X})$

Reconstructed image

$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{X}}[||\mathcal{F}(\mathbf{X}) - \mathbf{X}||]$$

Source: Isola, Freeman, Torralba



Source: Isola, Freeman, Torralba

Autoencoder

Data

$\{\mathbf{x}_i\}_{i=1}^{N} \quad \rightarrow$

Learner

Objective

$$\mathcal{L}(f(\mathbf{x}), \mathbf{x}) = \|f(\mathbf{x}) - \mathbf{x}\|_2^2$$

Hypothesis space

Neural net with a bottleneck

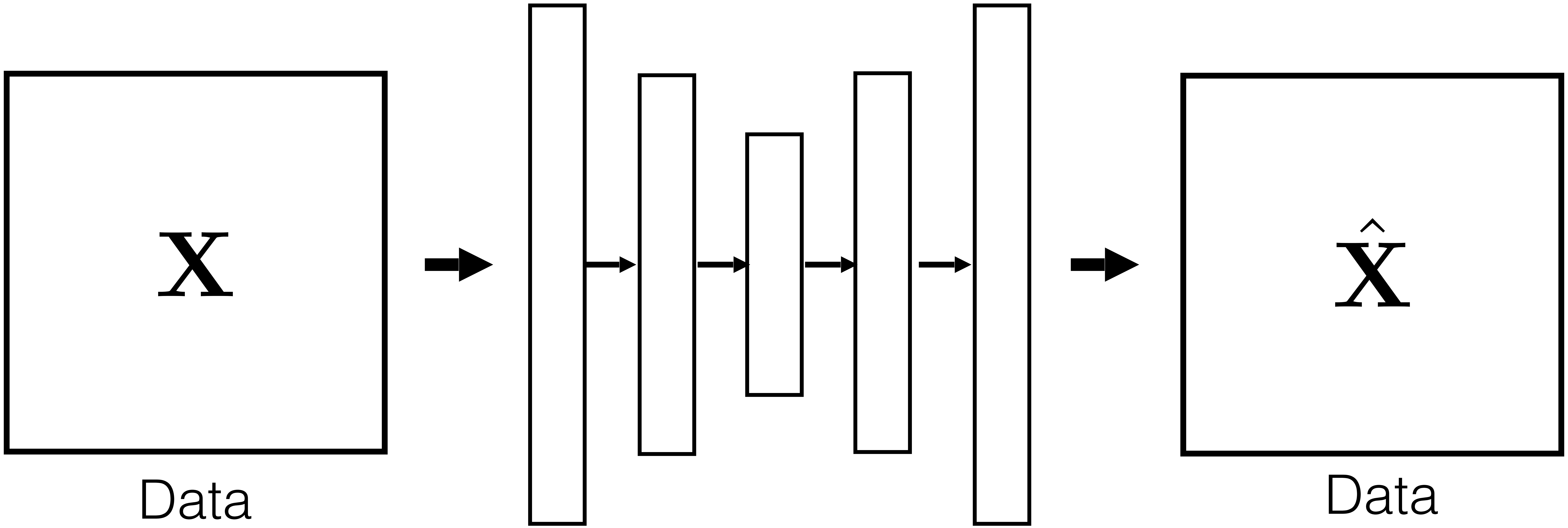
Optimizer

SGD

$\rightarrow f$

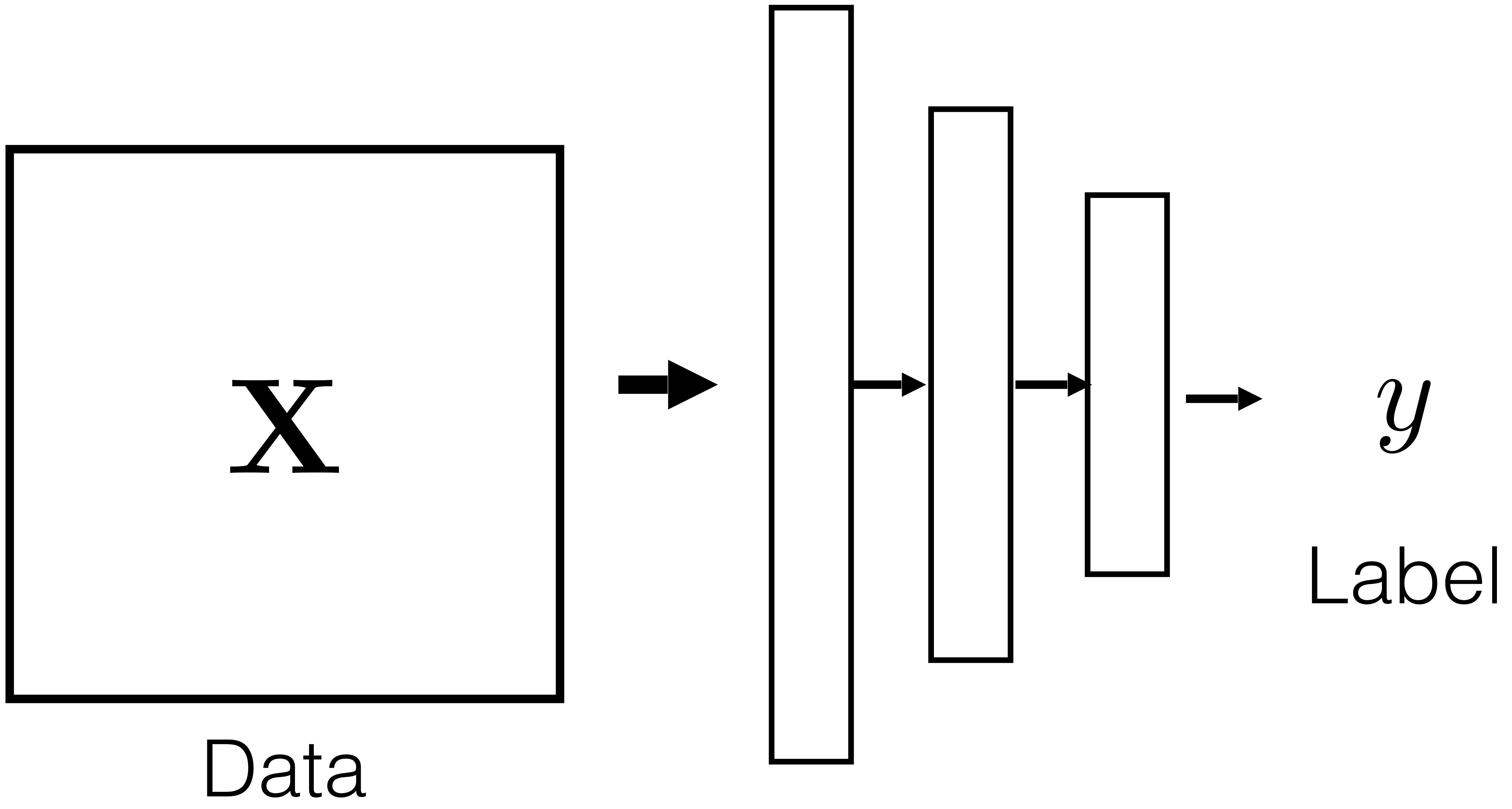
Source: Isola, Freeman, Torralba

Data compression



Source: Isola, Freeman, Torralba

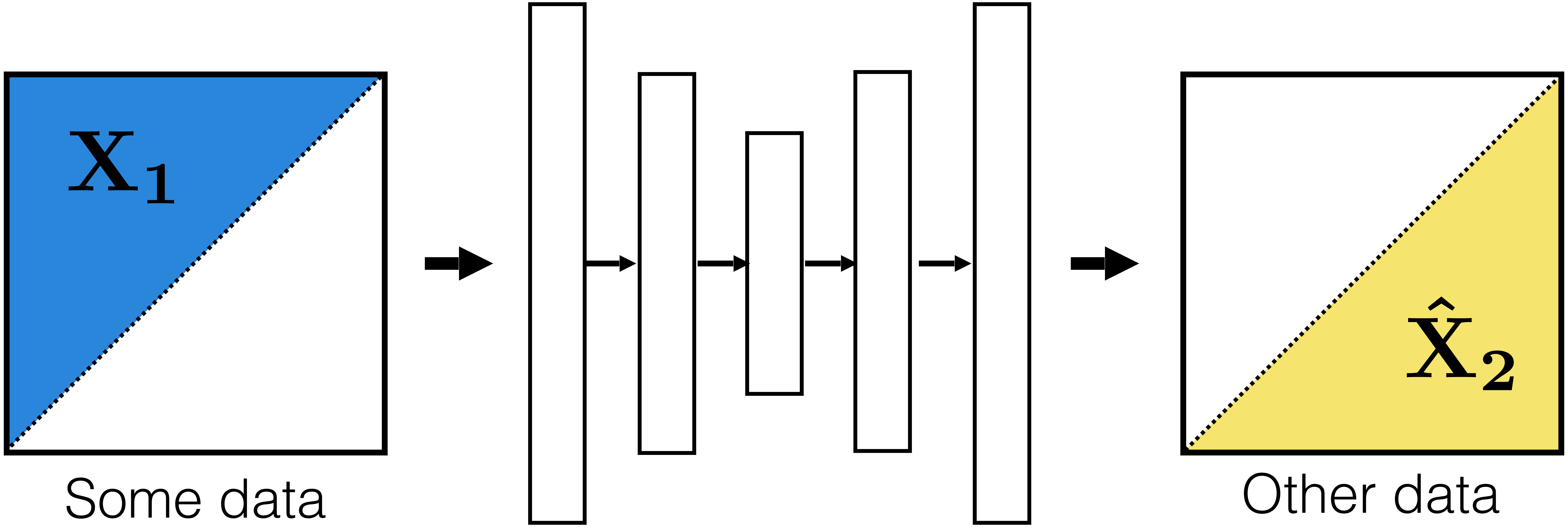
Label prediction



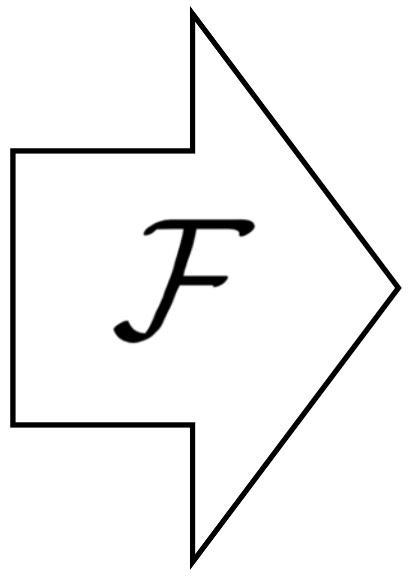
e.g., image classification

Source: Isola, Freeman, Torralba

Data prediction
aka "self-supervised learning"



Source: Isola, Freeman, Torralba

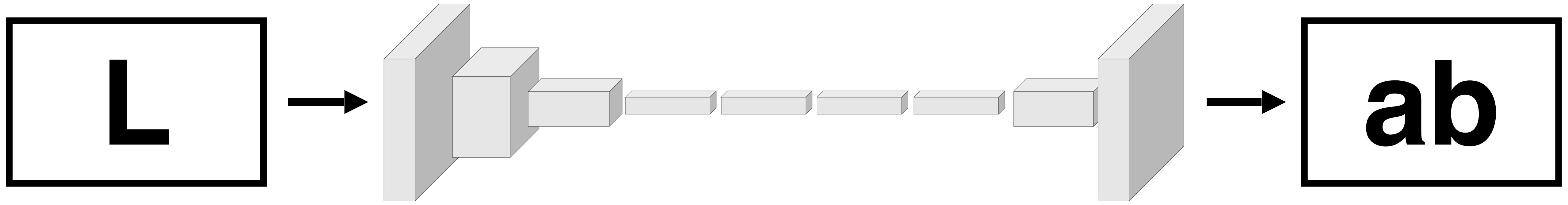


Grayscale image: L channel

$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Color information: ab channels

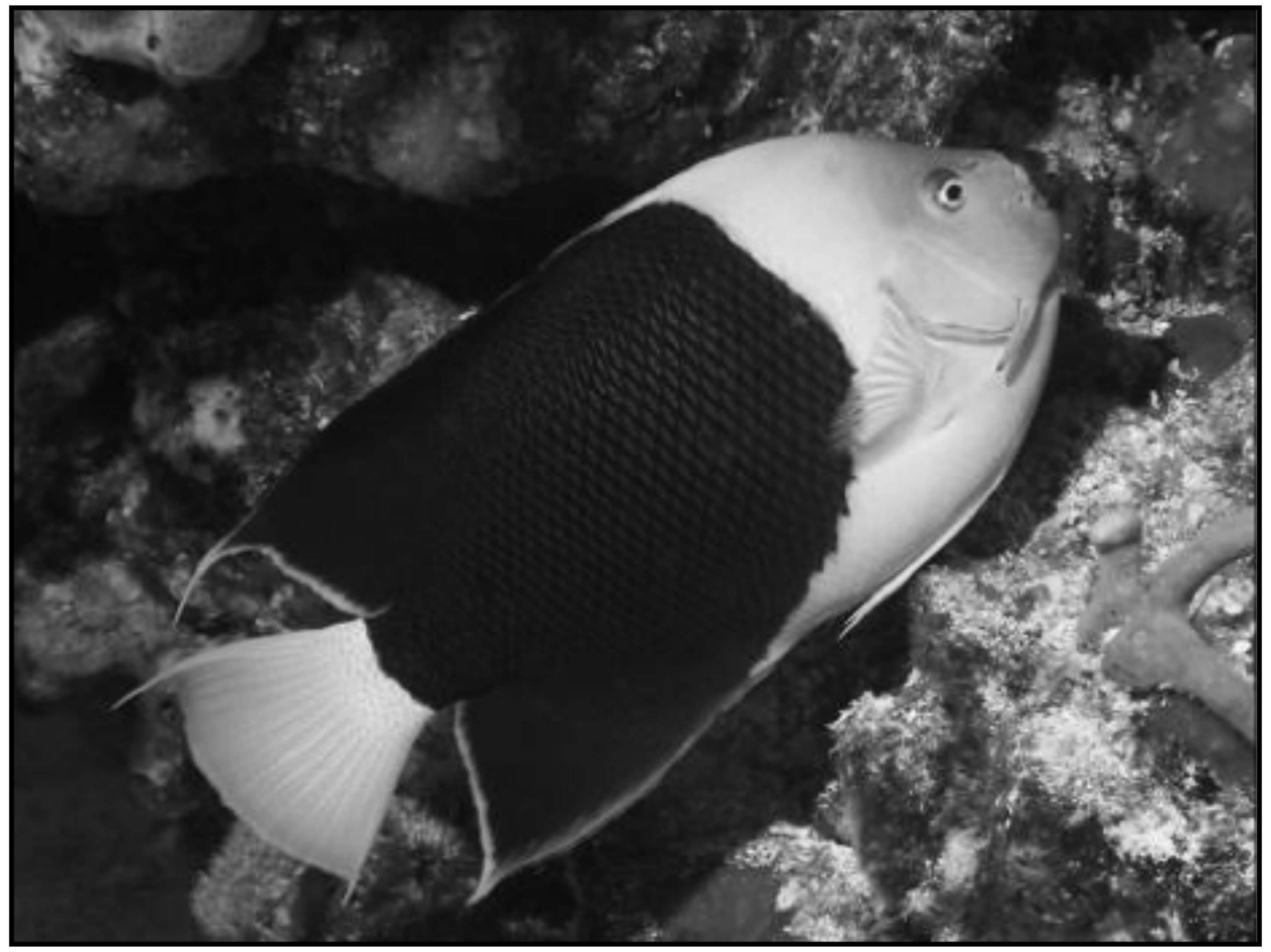
$\widehat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$

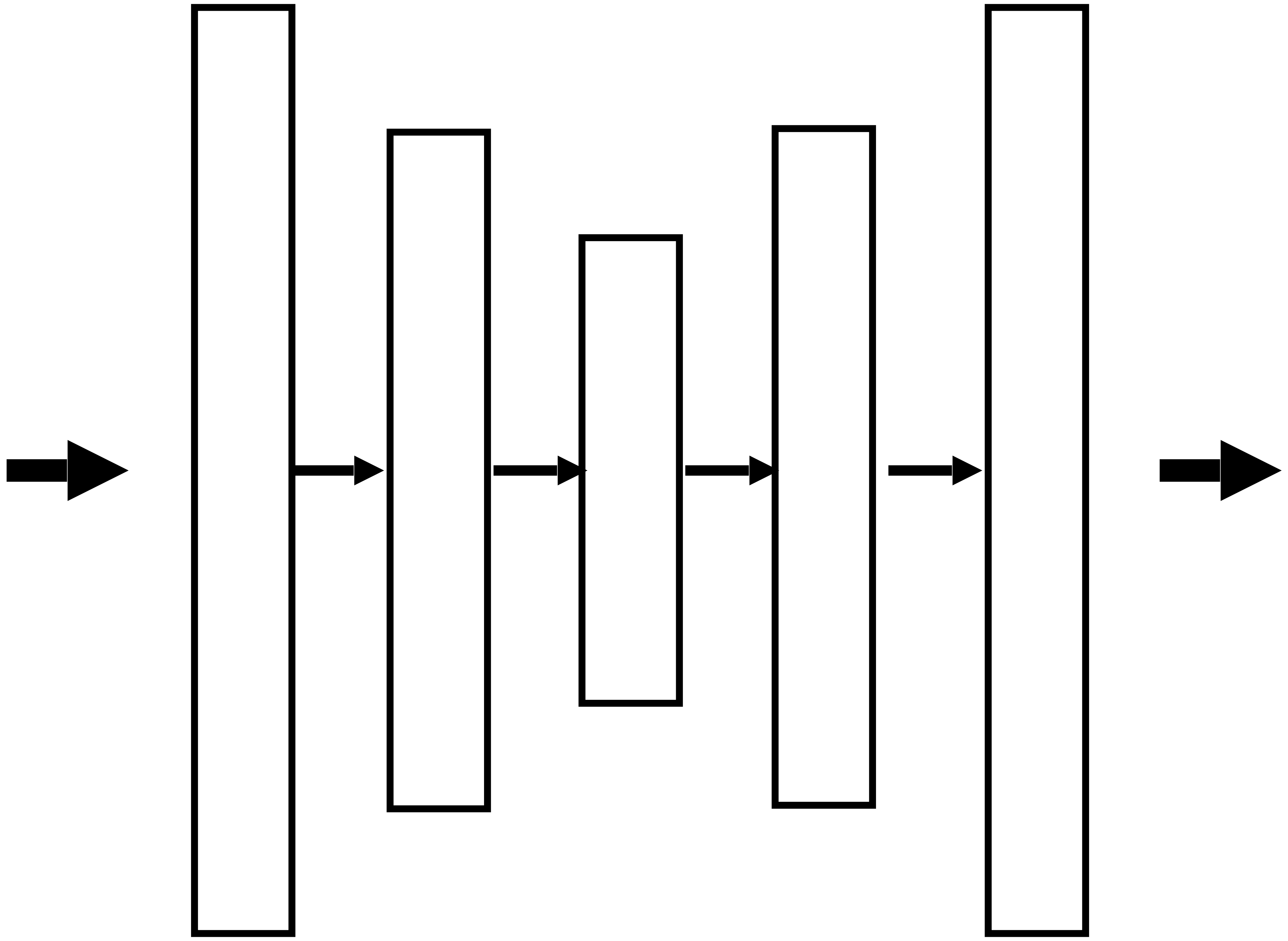


[Zhang, Isola, Efros, ECCV 2016]

Source: Isola, Freeman, Torralba

Visualizing units





[Zeiler & Fergus, ECCV 2014]

[Zhou et al., ICLR 2015]

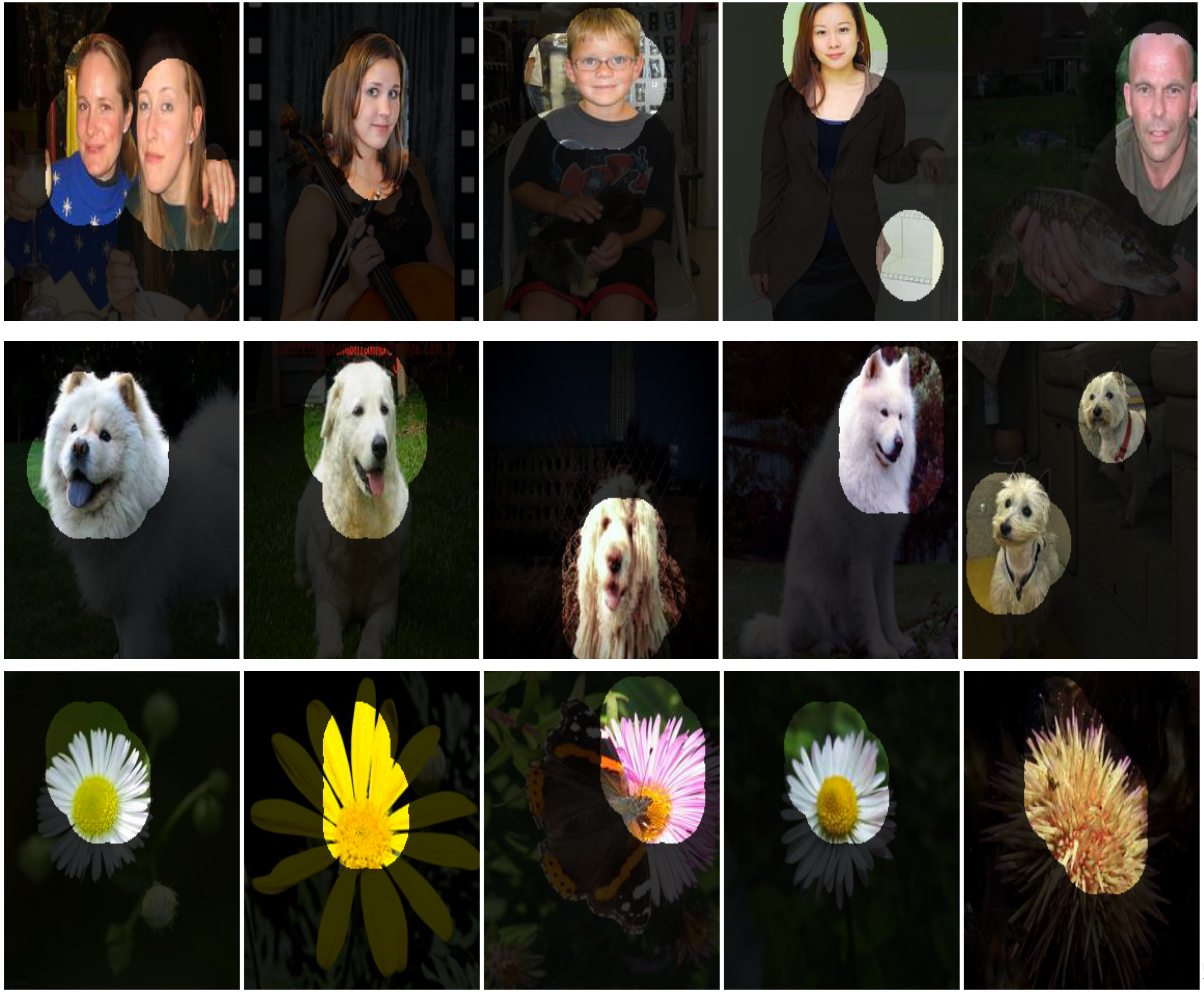
Source: Isola, Freeman, Torralba

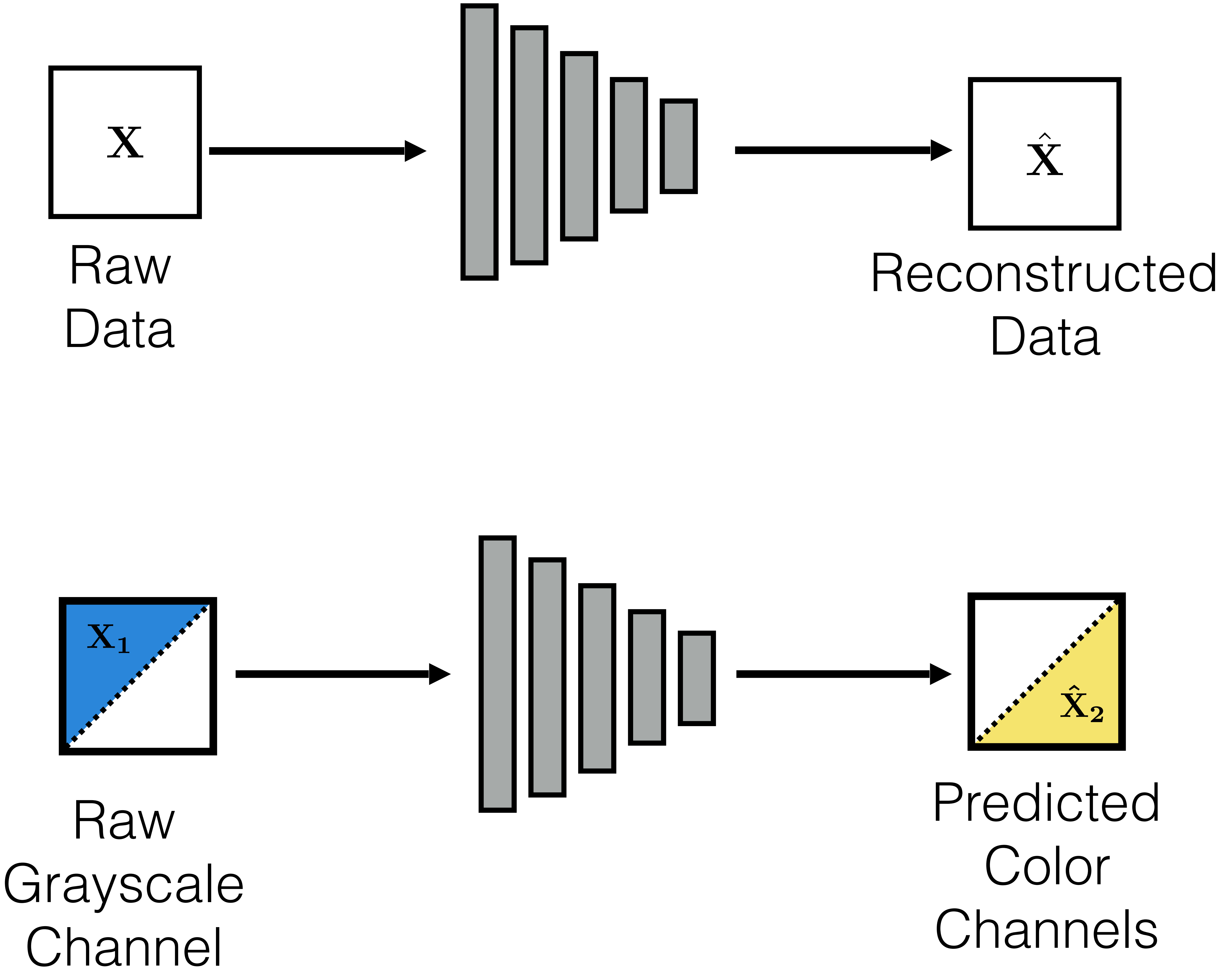
Stimuli that drive selected neurons (conv5 layer)

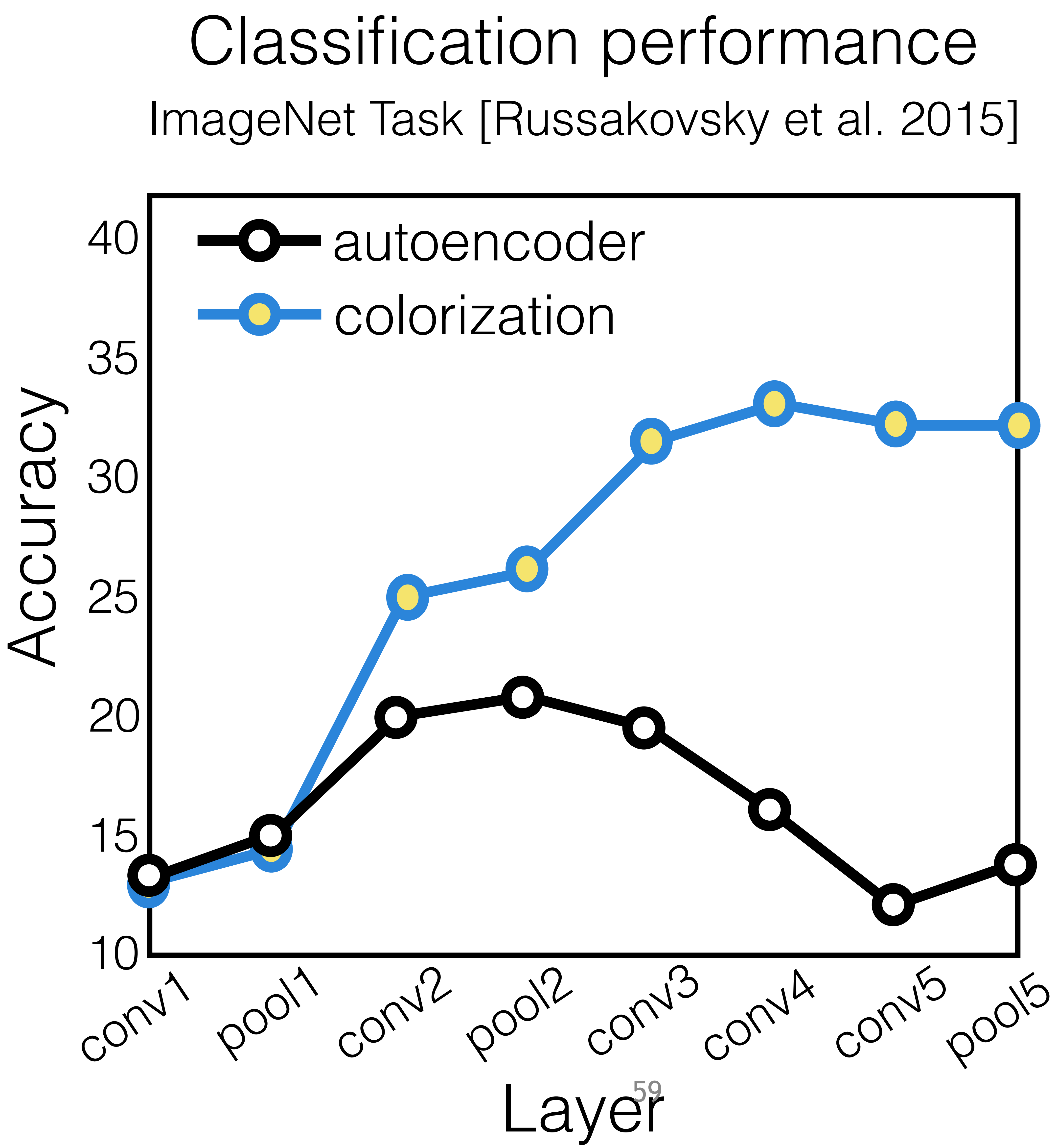
faces

dog faces

flowers





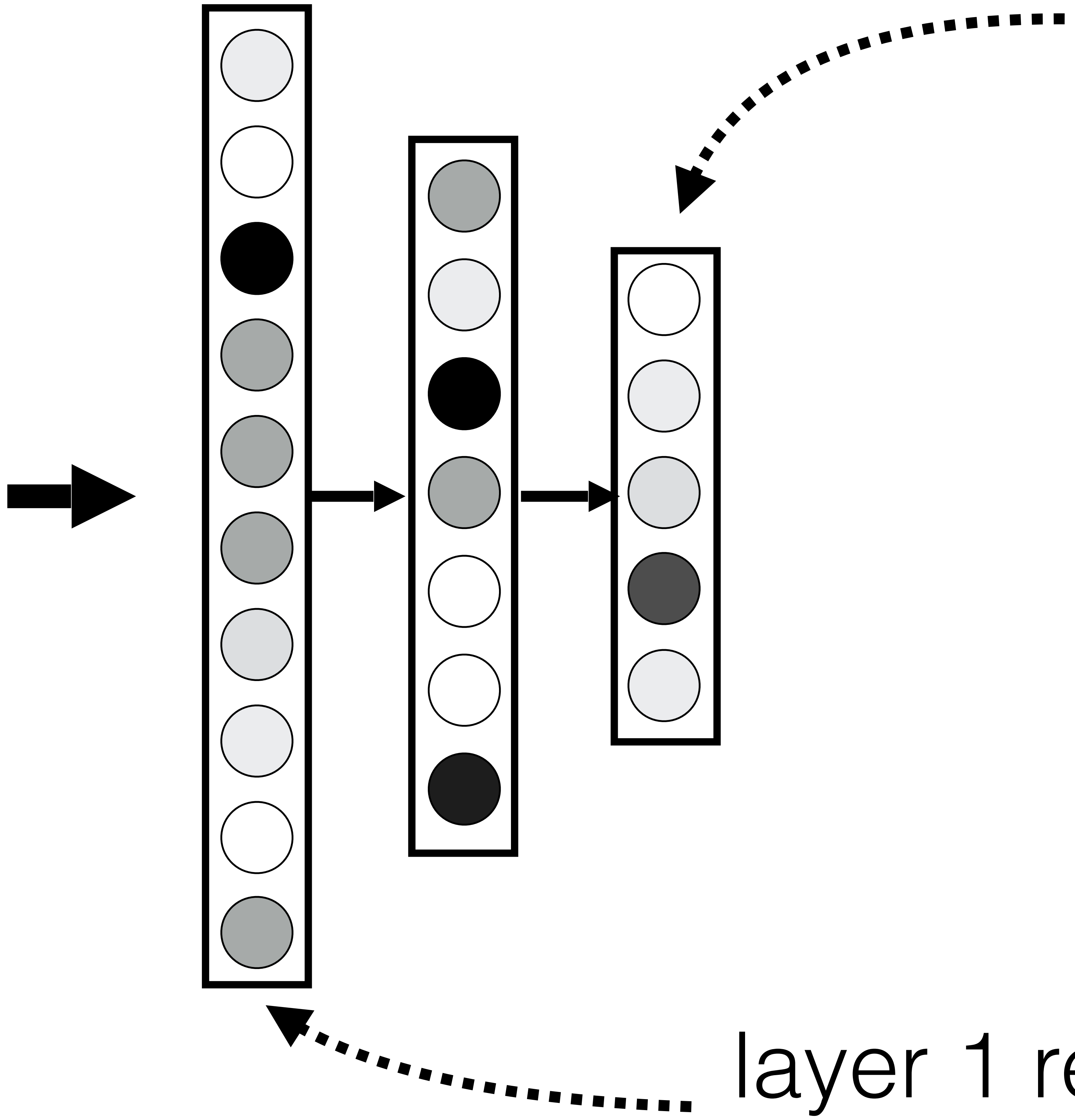


Source: Isola, Freeman, Torralba

im2vec

$\mathbf{x}$

Image



layer 3 representation of image

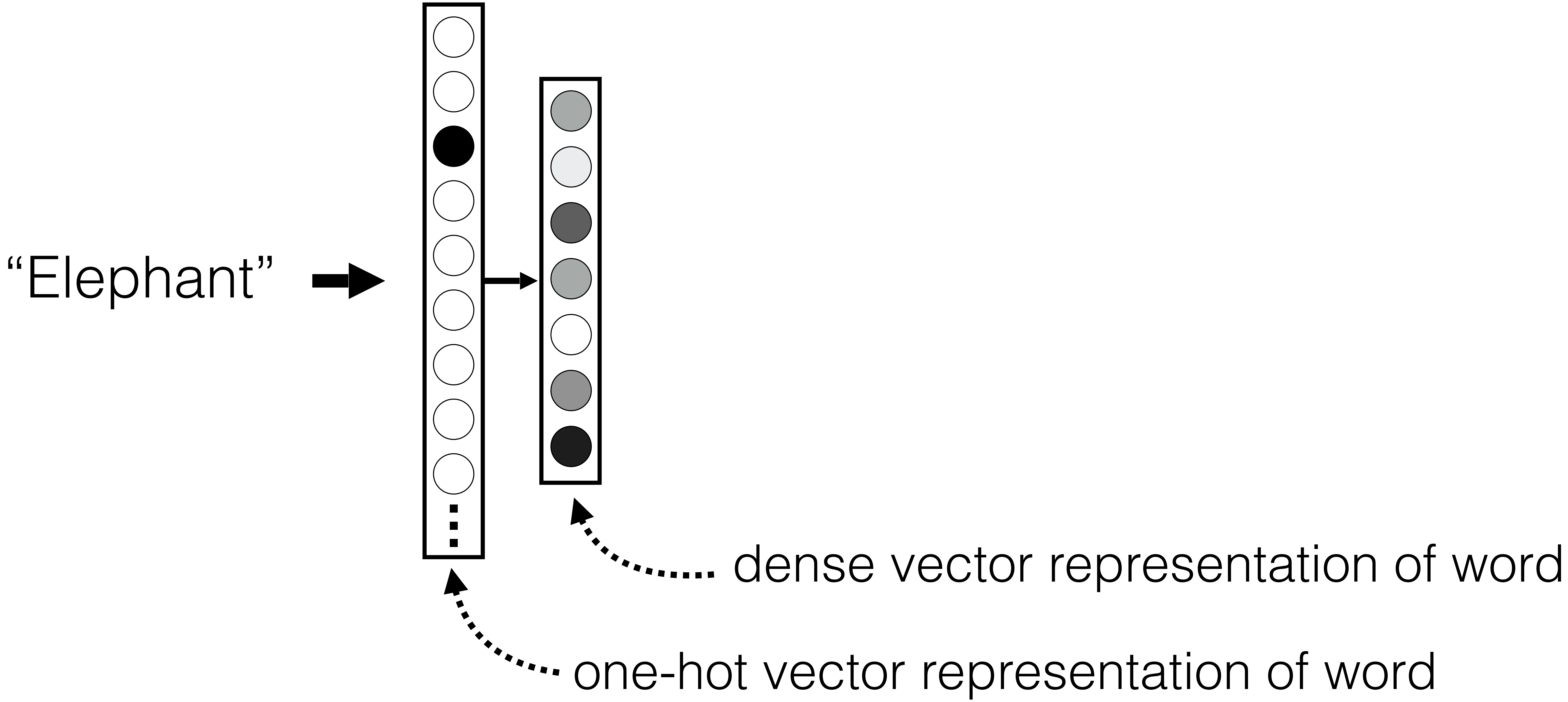
layer 1 representation of image

Represent image as a vector of neural activations

(perhaps representing a vector of detected texture patterns or object parts)

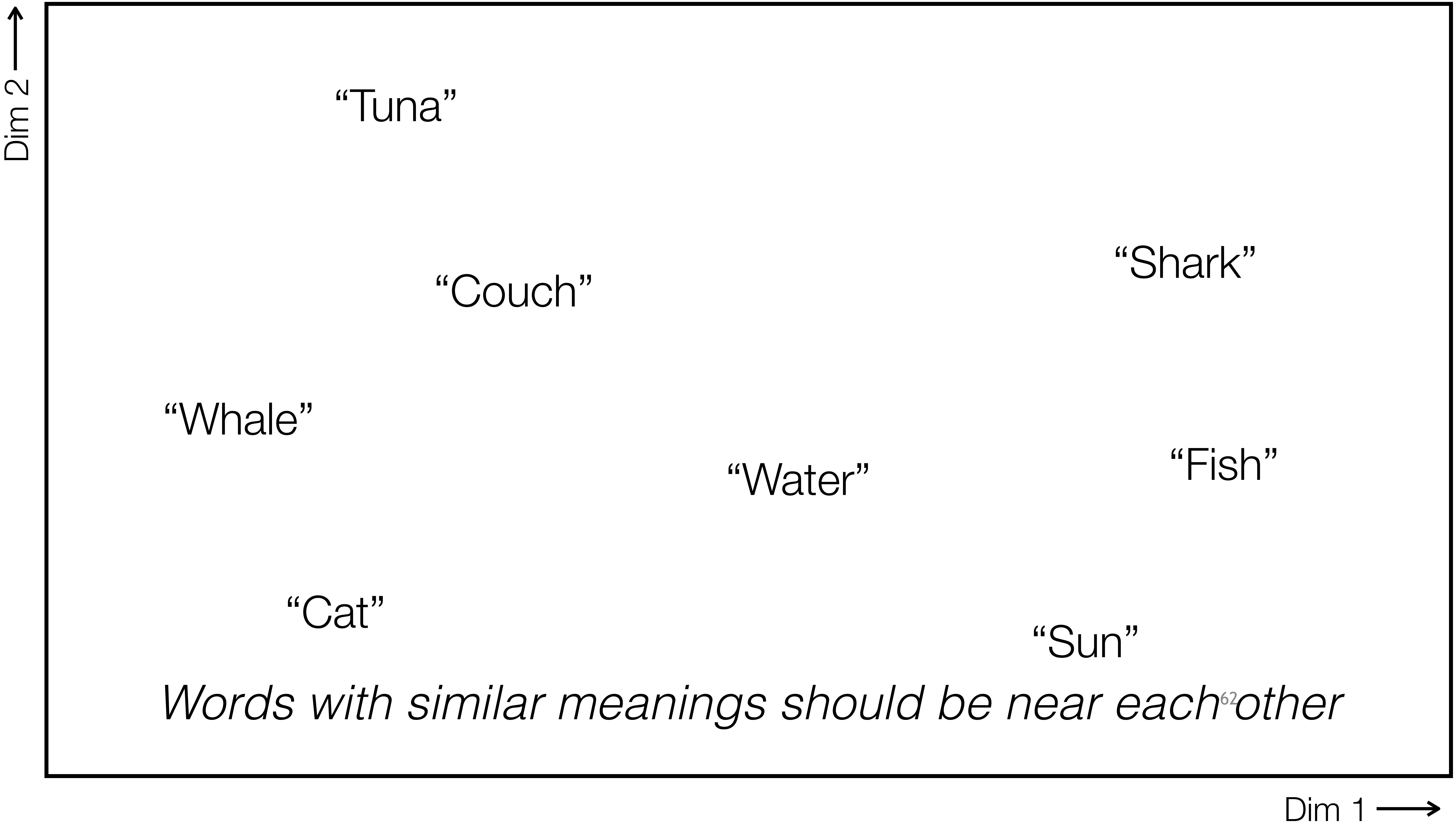
Source: Isola, Freeman, Torralba

word2vec



X2vec methods are also called embeddings of X, e.g., a **word embedding**

Source: Isola, Freeman, Torralba



Words with similar meanings should be near each other

word2vec

Words with similar meanings should be near each other

Proxy: words that are used in the same context tend to have similar meanings

words with similar contexts should be near each other

Source: Isola, Freeman, Torralba

Next to the 'sofa' is a desk, and a 'person' is sitting behind it.

'armchair'	'man'
'bench'	'woman'
'chair'	'child'
'deck chair'	'teenager'
'ottoman'	'girl'
'seat'	'boy'
'stool'	'baby'
'swivel chair'	'daughter'
'loveseat'	'son'
…	…

word2vec

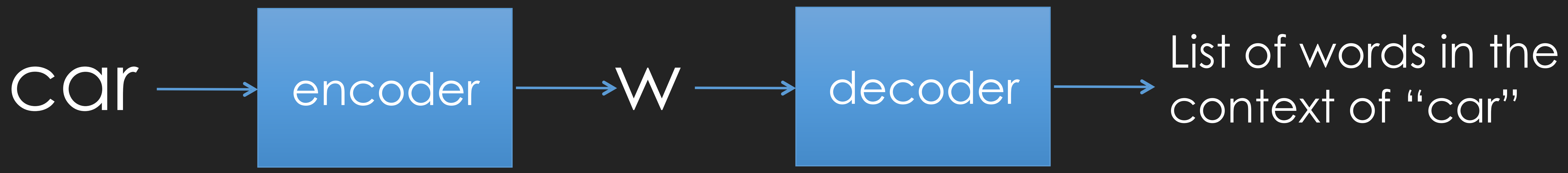
I parked the car in a nearby street. It is a red car with two doors, …

I parked the vehicle in a nearby street…

T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781, 2013

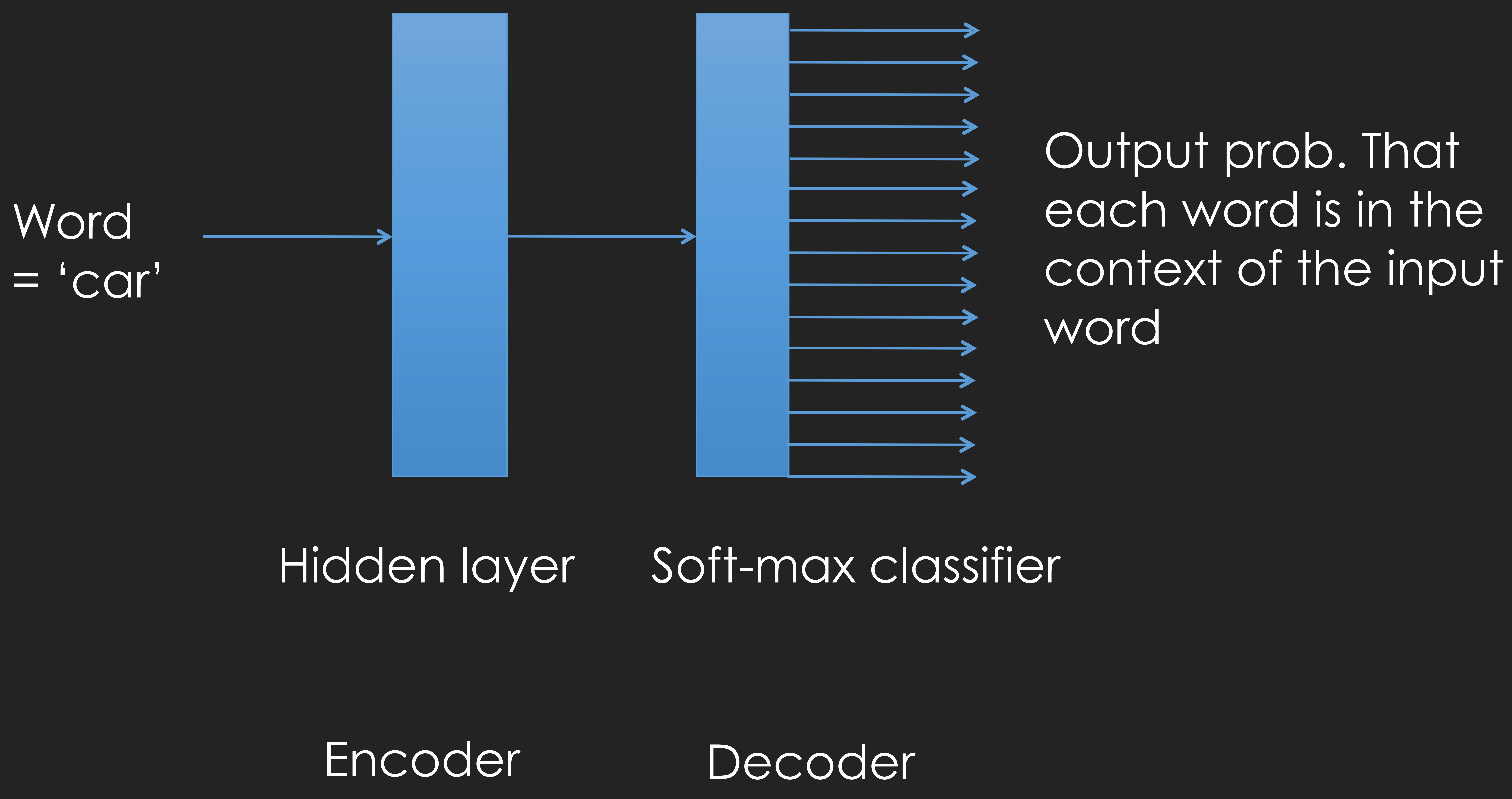
word2vec

I parked the car in a nearby street. It is a red car with two doors, …



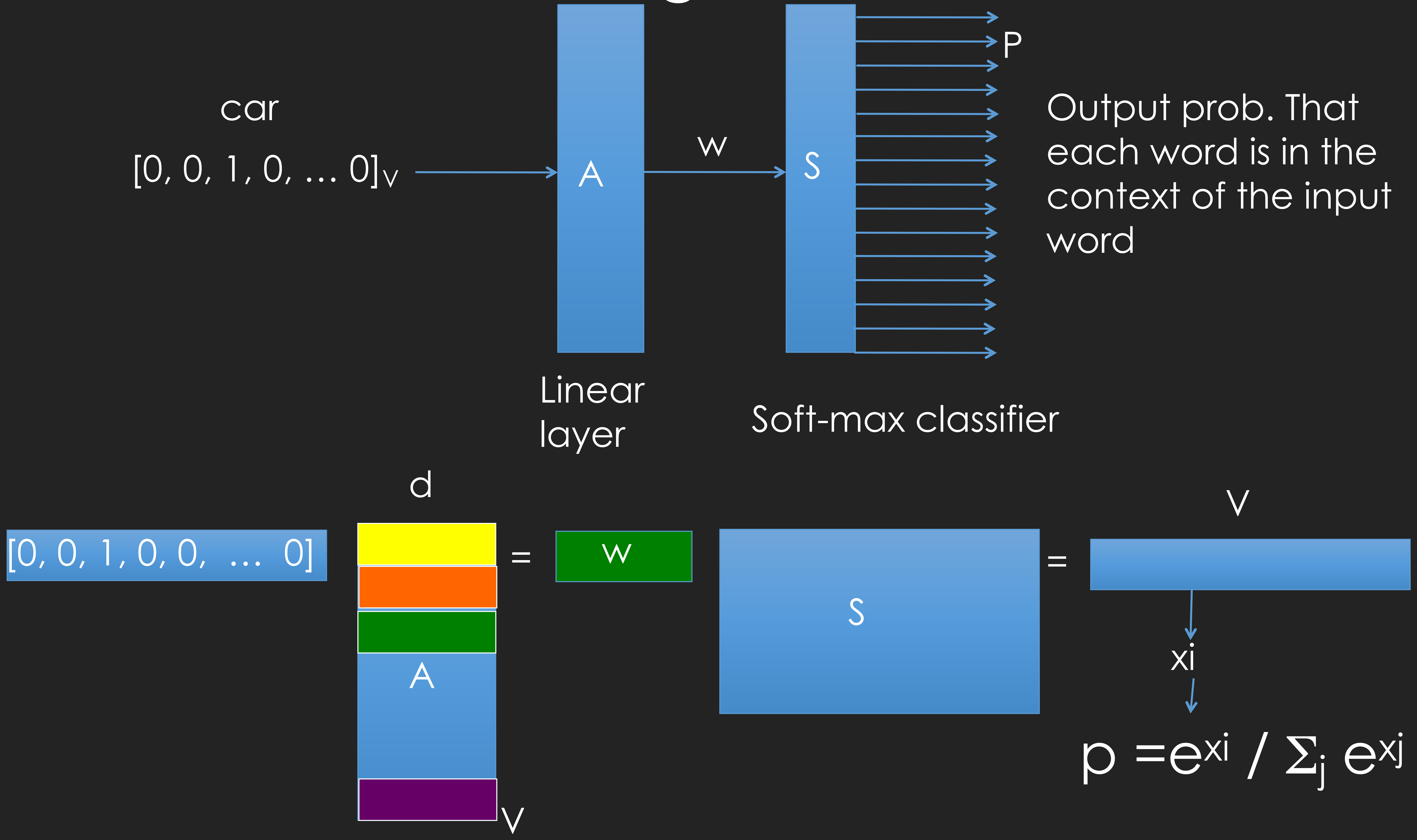
T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781, 2013

word2vec



T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781, 2013

word2vec, training



T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781, 2013

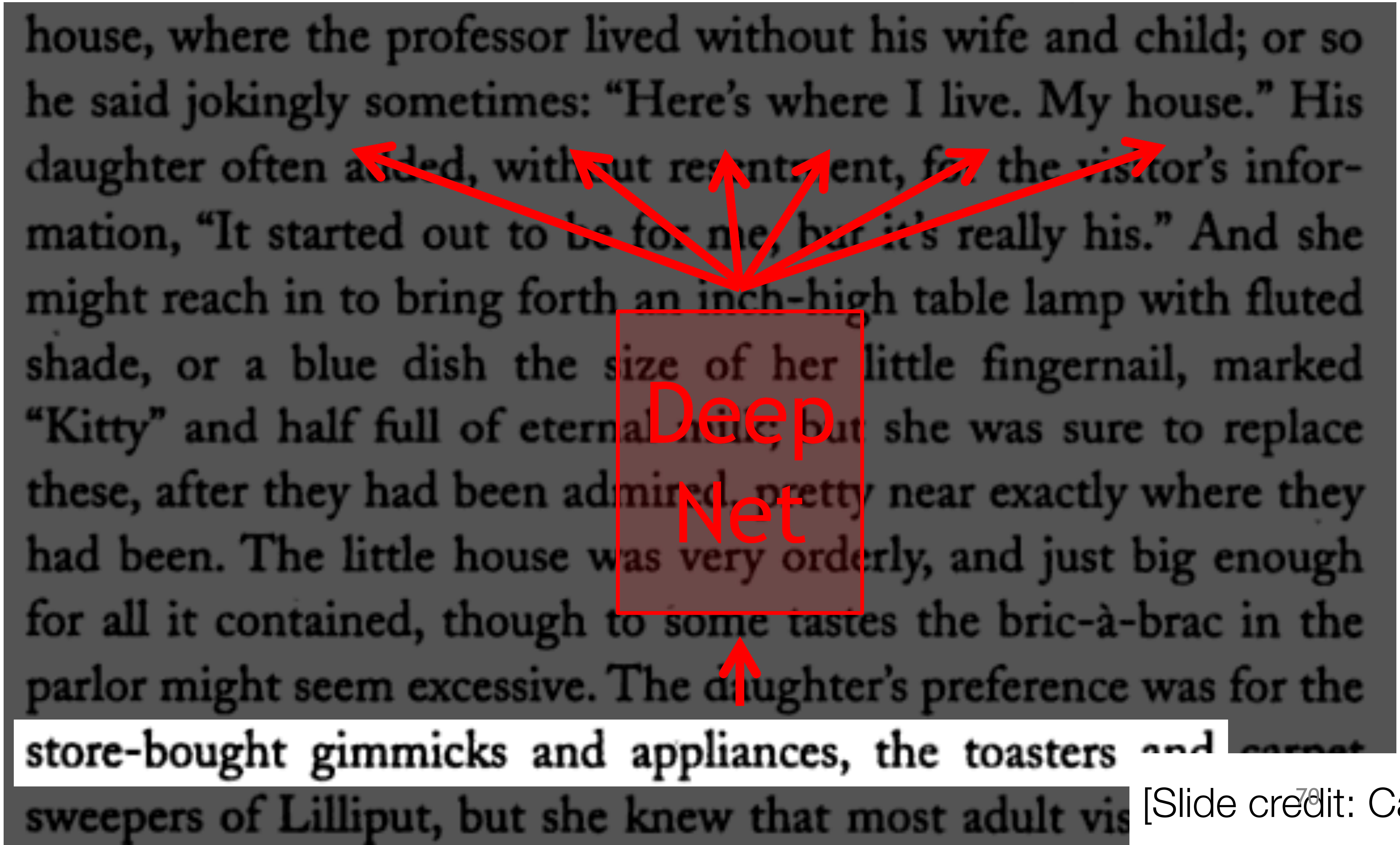
Algebraic operations with the vector representation of words

X = Vector(“Paris”) – vector(“France”) + vector(“Italy”)

Closest nearest neighbor to X is vector(“Rome”)

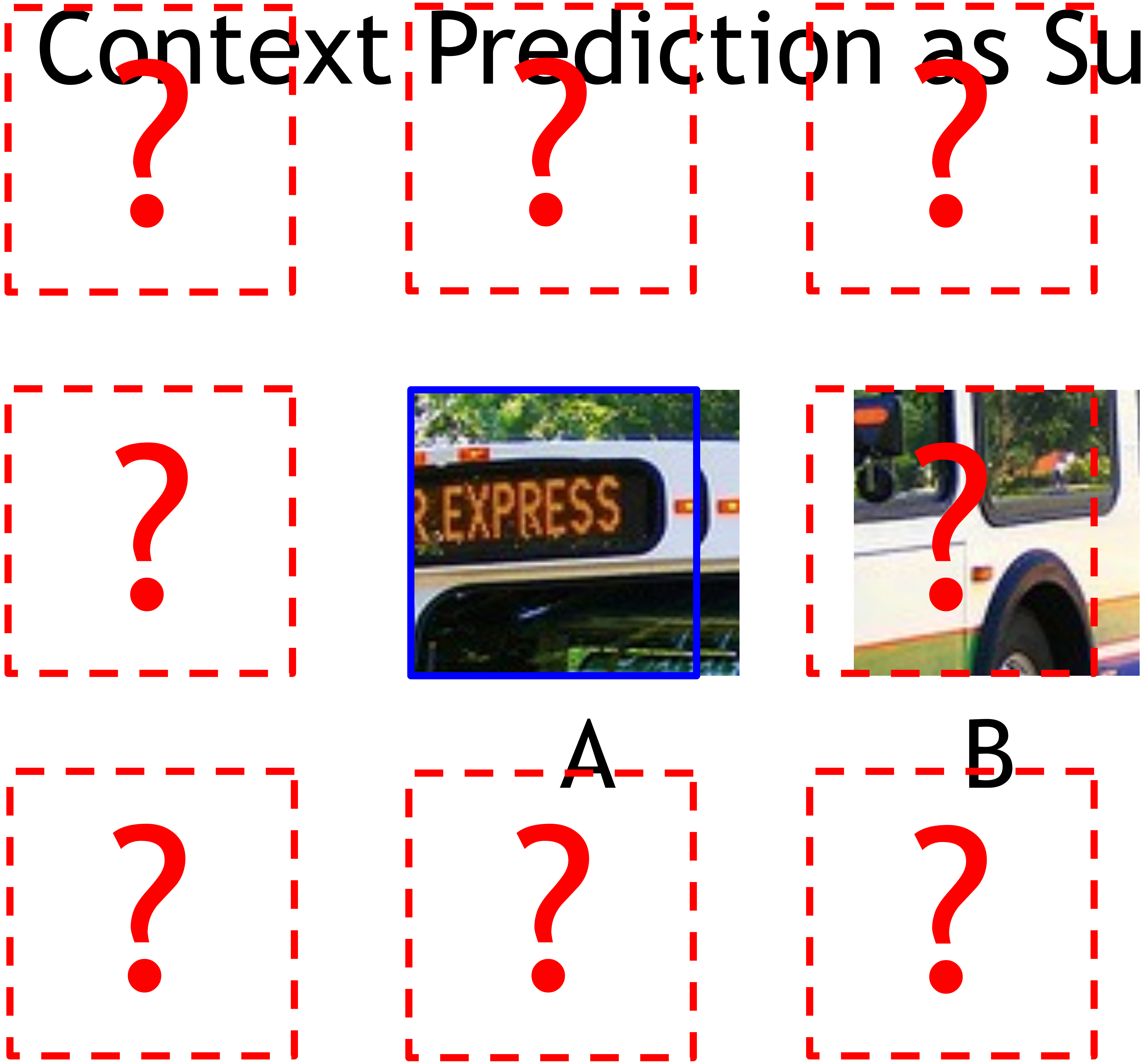
Context as Supervision

[Collobert & Weston 2008; Mikolov et al. 2013]



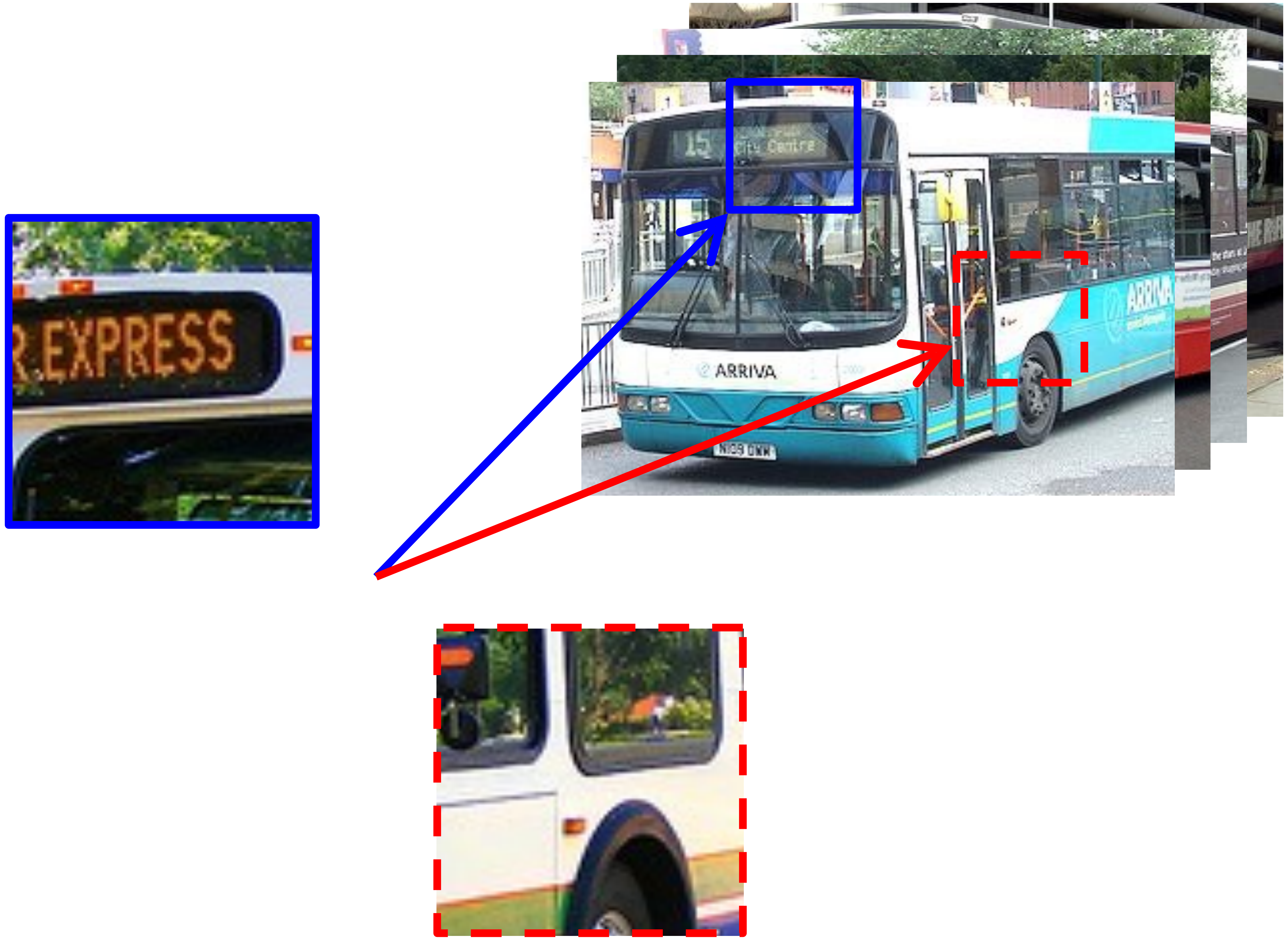
[Slide credit: Carl Doersch]

Context Prediction as Supervision



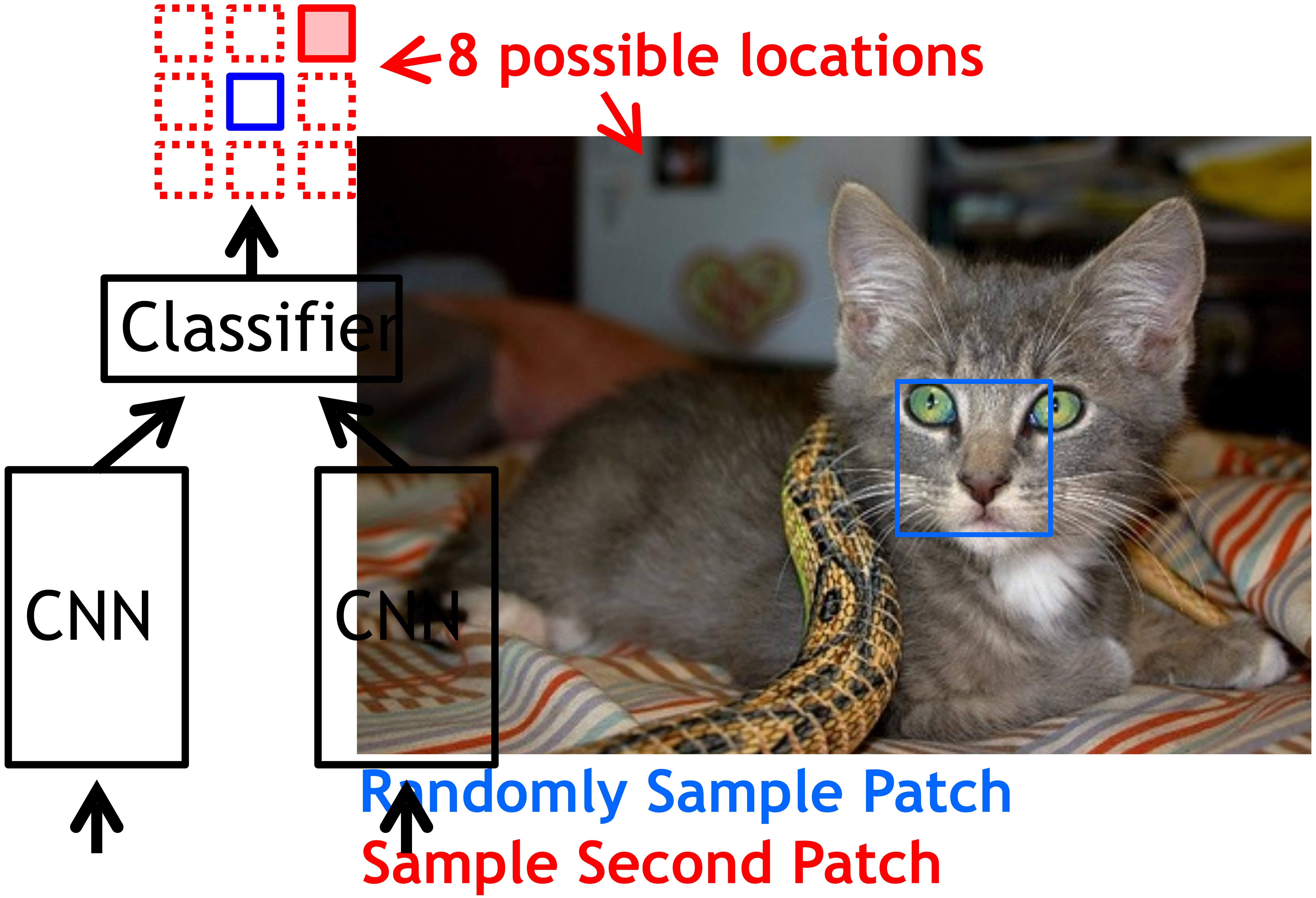
[Slide credit: Carl Doersch]

Semantics from a non-semantic task

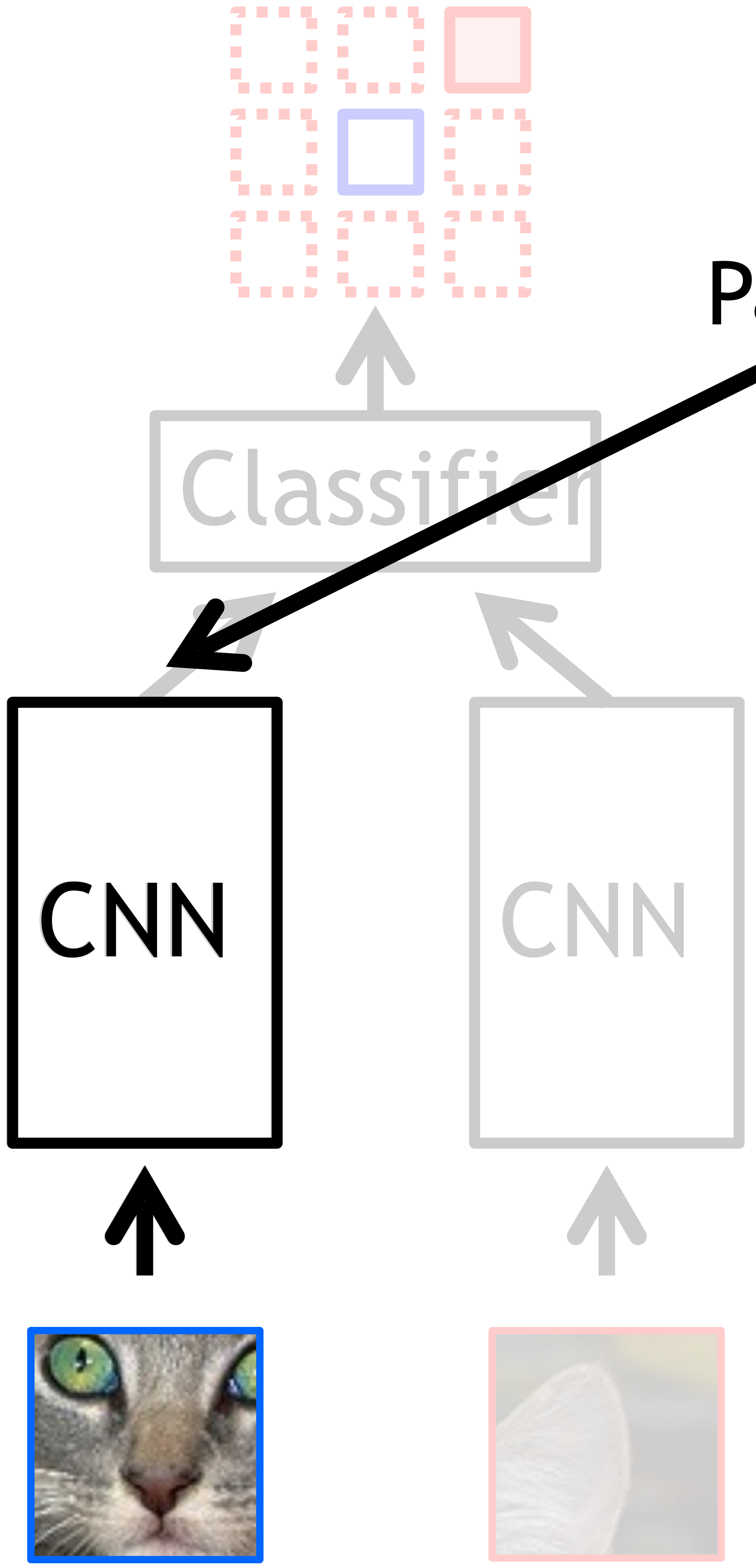


[Slide credit: Carl Doersch]

Relative Position Task



[Slide credit: Carl Doersch]



Patch Embedding (representation)

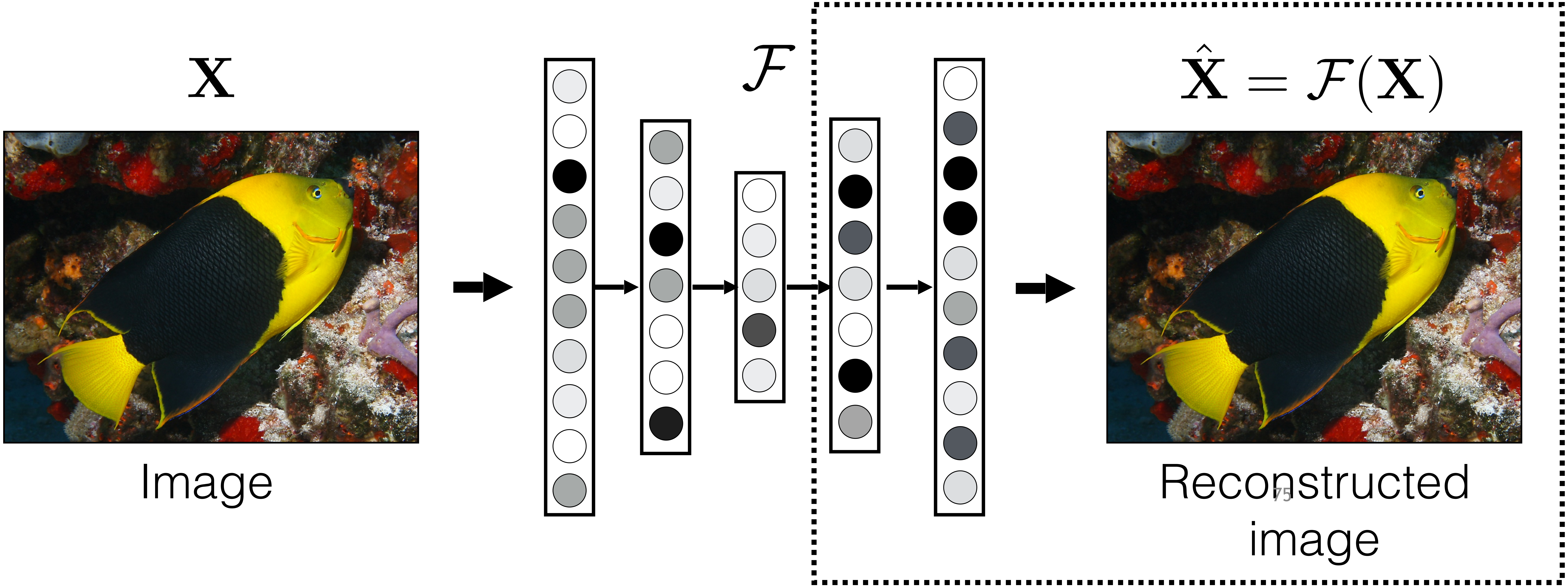
Input   Nearest Neighbors

Note: connects ***across*** instances!

[Slide credit: Carl Doersch]

Revisiting autoencoders

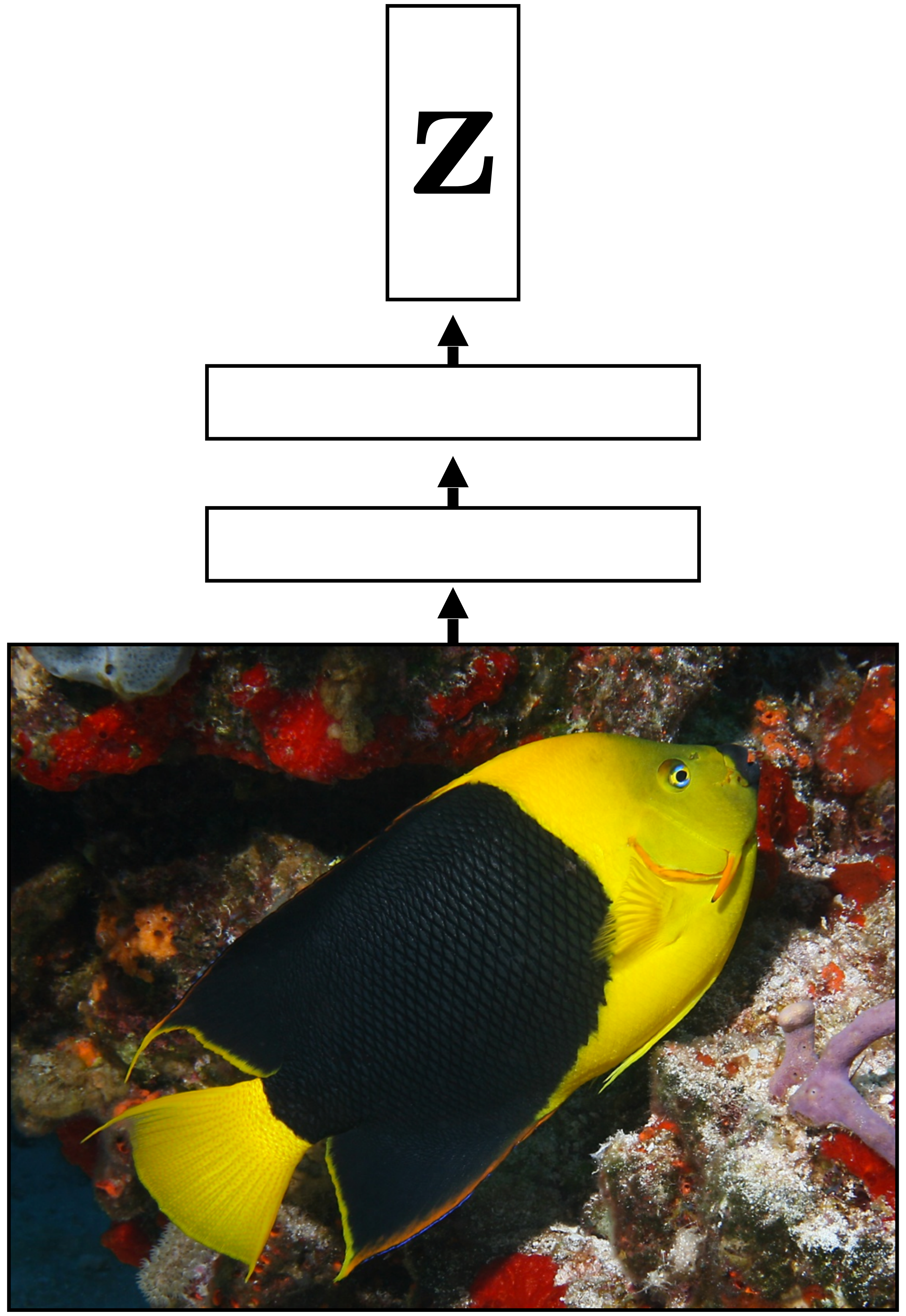
Is reconstruction necessary?

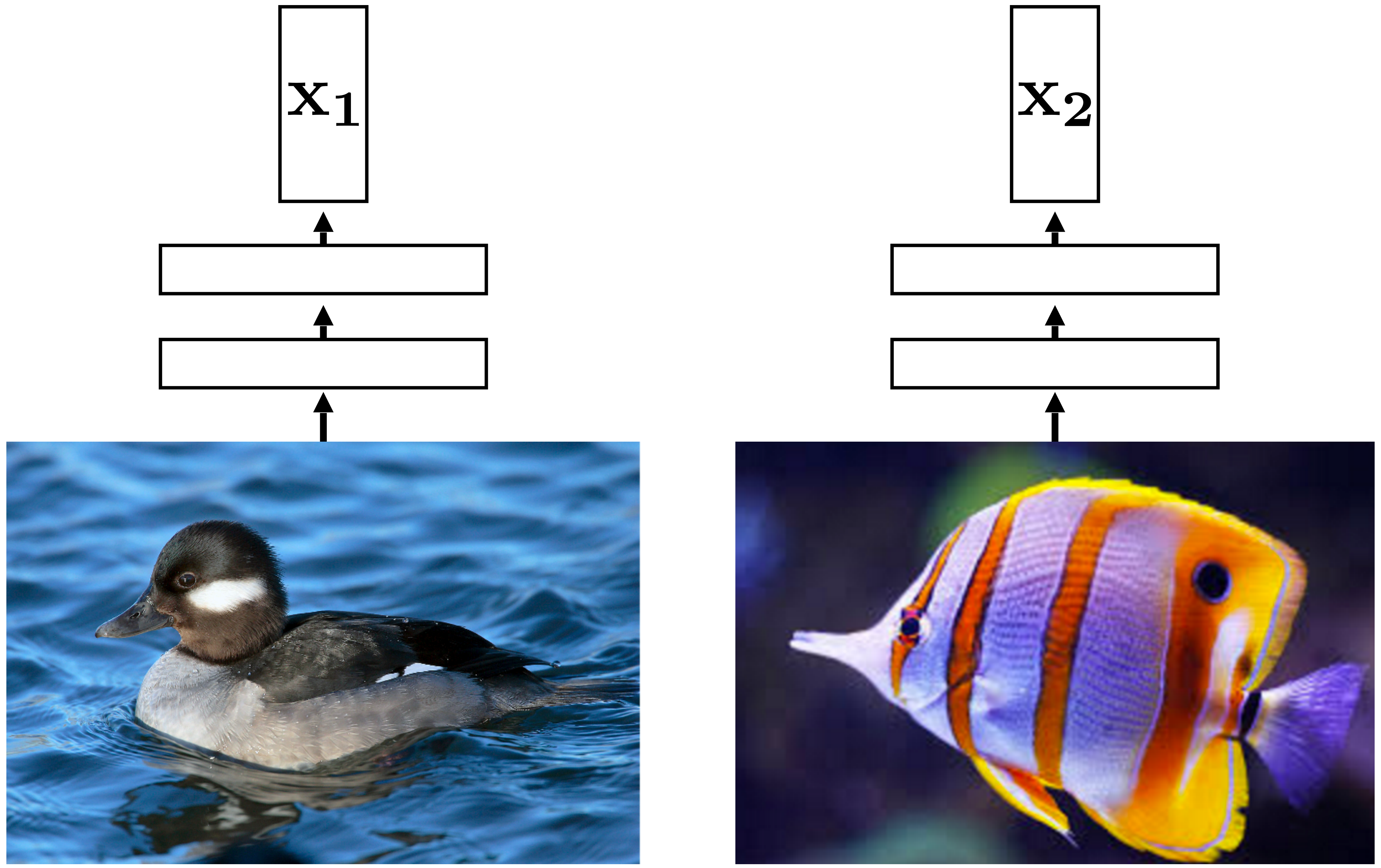


Contrastive learning

$$\mathbf{z}^\top \mathbf{z} \longrightarrow \text{High}$$

$$\mathbf{z}^\top \mathbf{x}_1 \longrightarrow \text{Low}$$

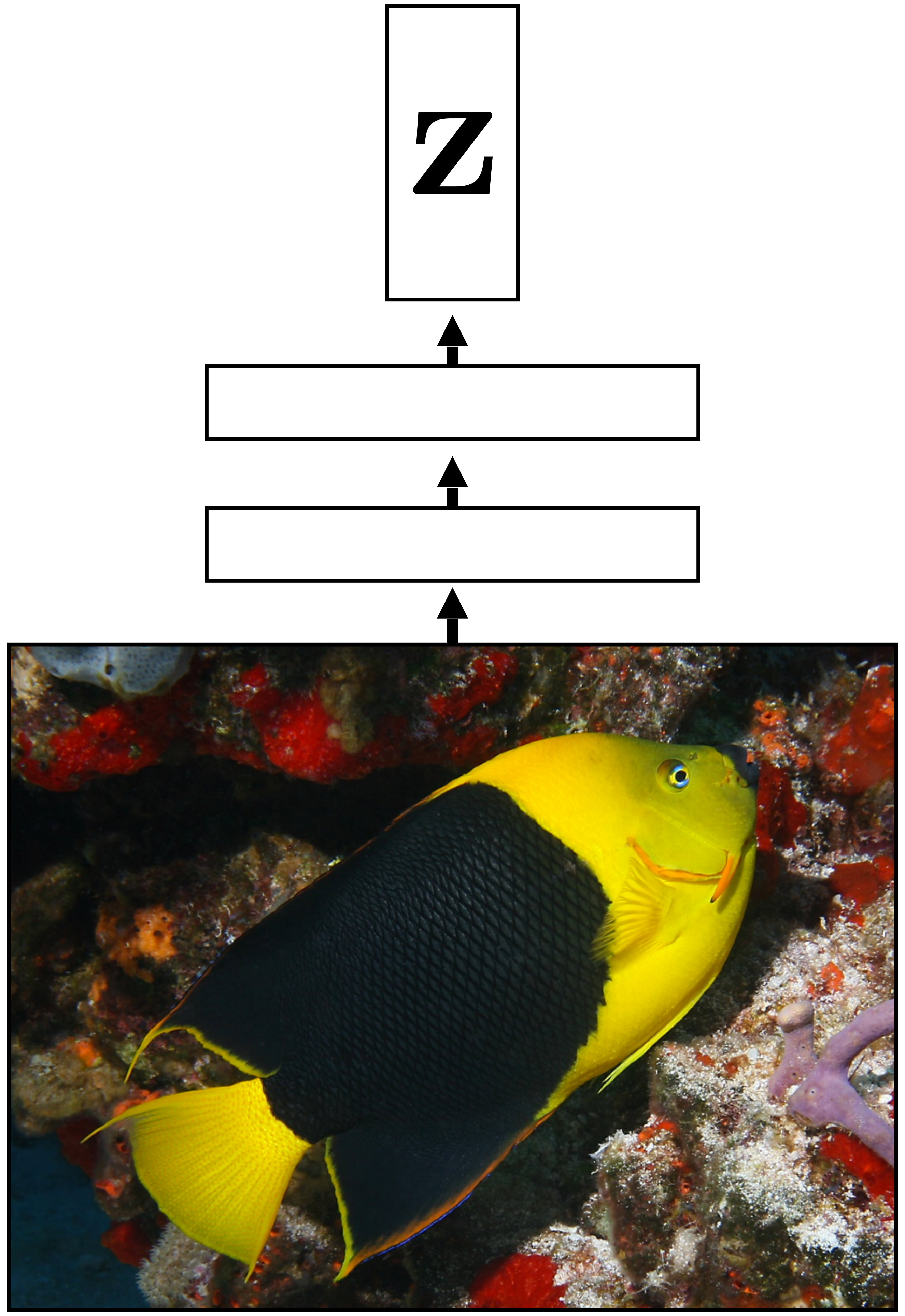




...

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]

Contrastive learning



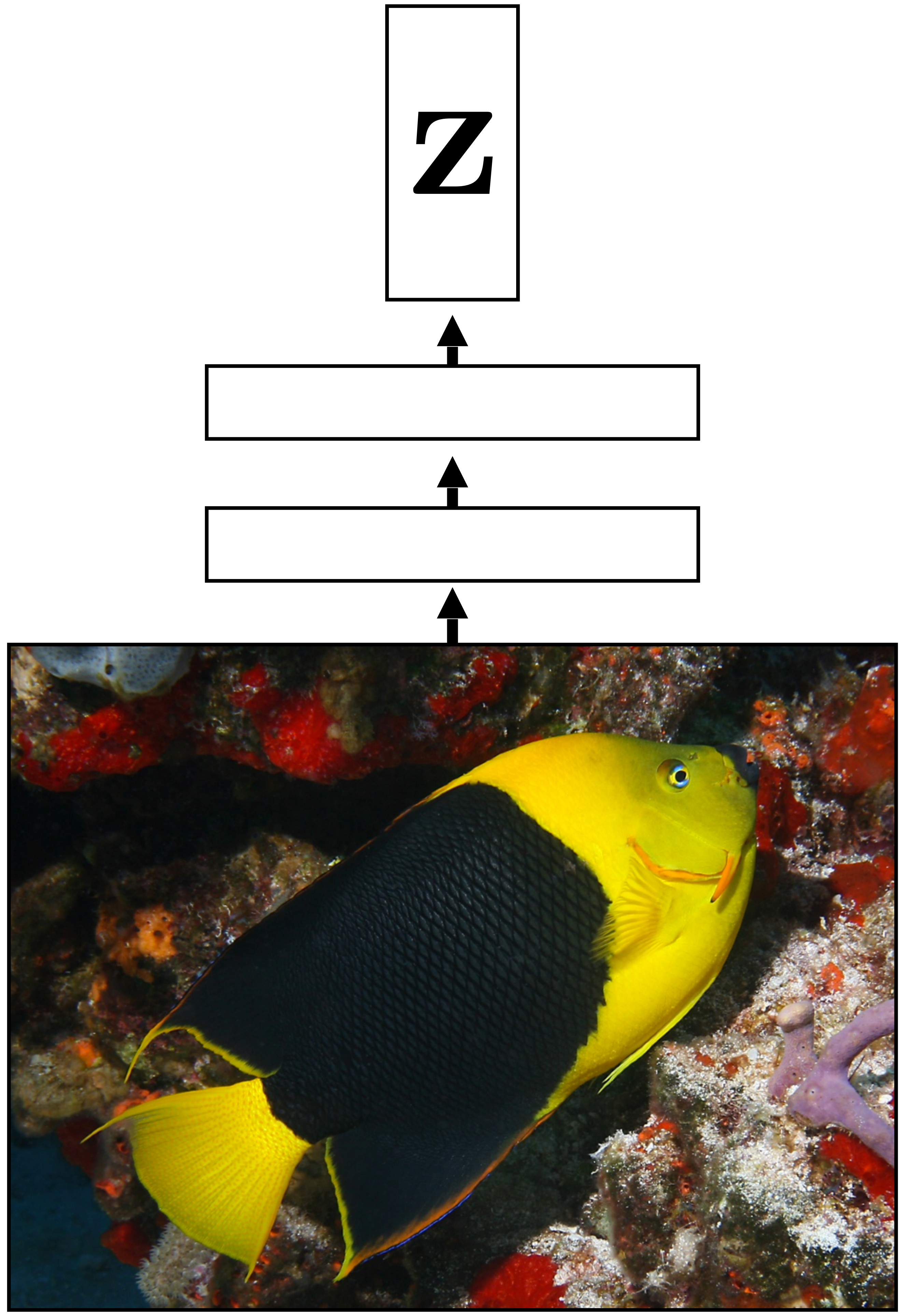
Maximize:

$$\frac{\exp\{\mathbf{z}^\top \mathbf{z}\}}{\exp\{\mathbf{z}^\top \mathbf{z} + \sum_i \mathbf{z}^\top \mathbf{x}_i\}}$$

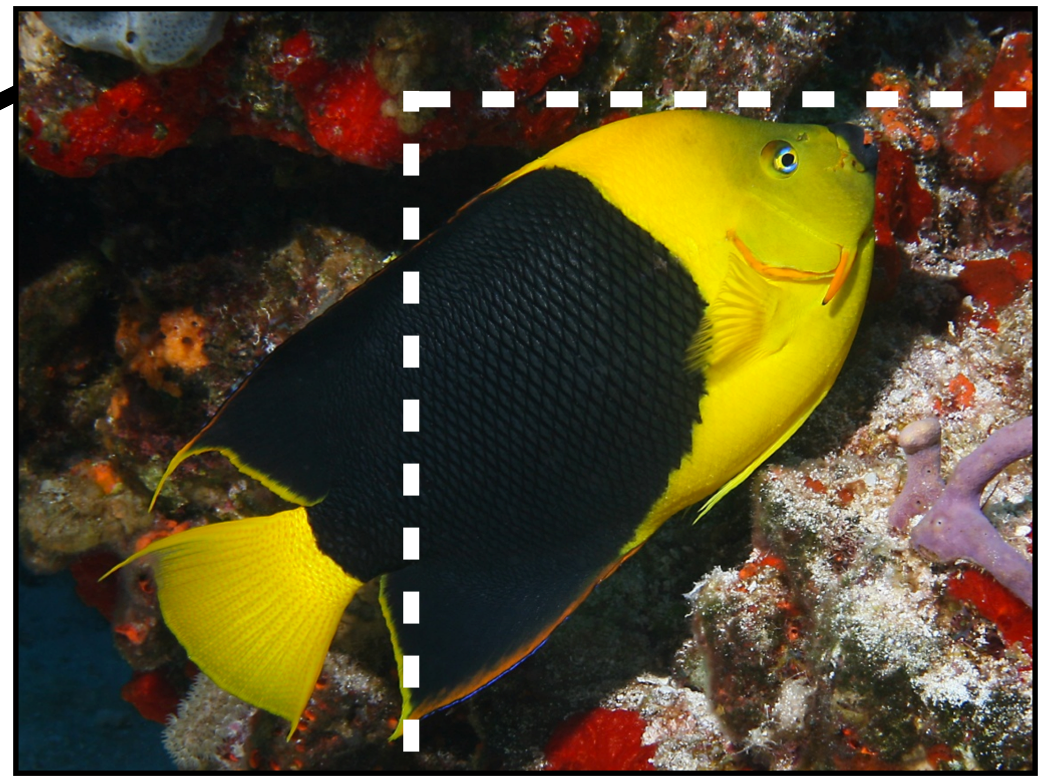
Equivalent to softmax loss with each image as a category

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]

Contrastive learning



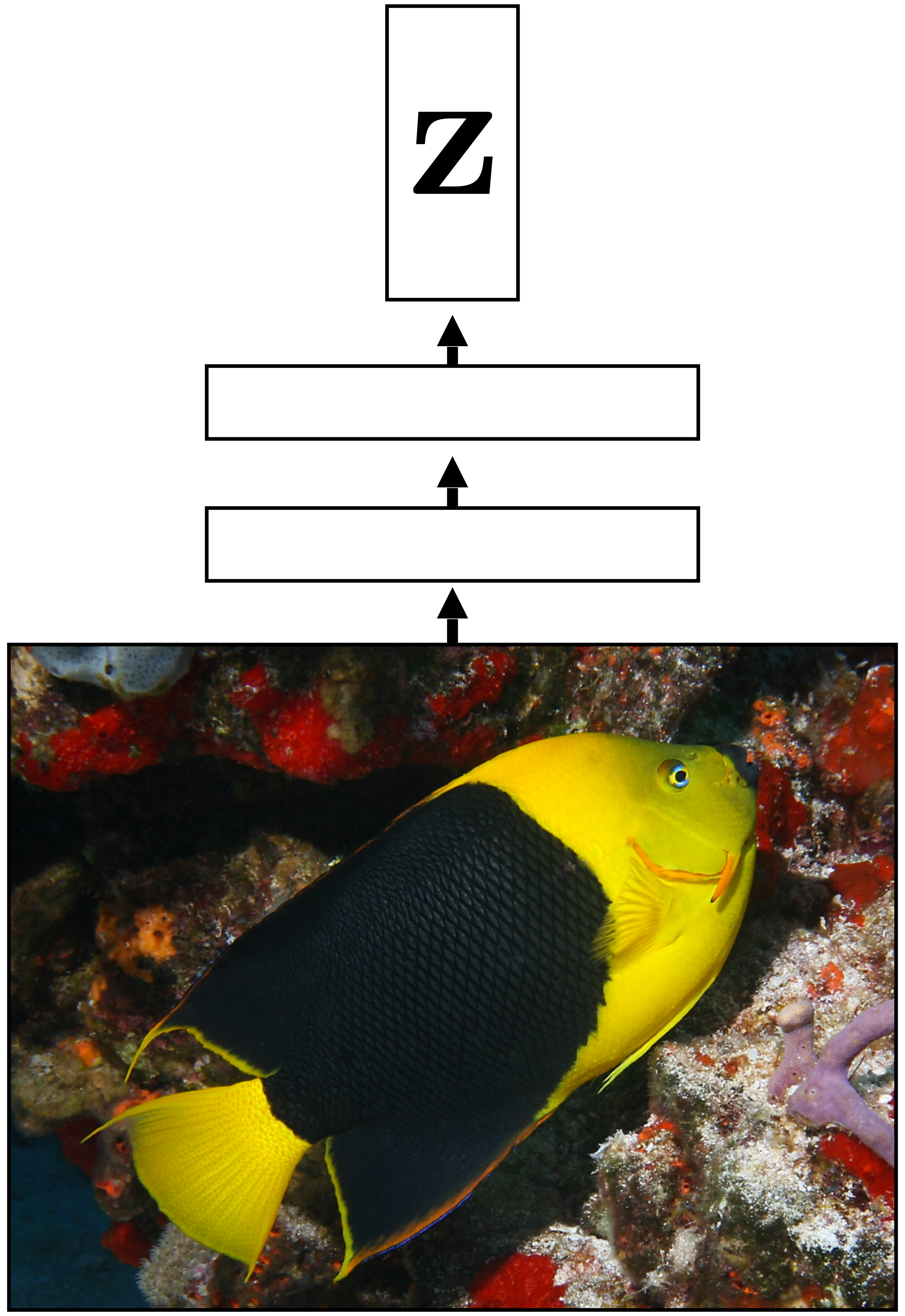
Can build invariance.
Compare to *warped* images.



$$\frac{\exp\{\mathbf{z}^\top \tilde{\mathbf{z}}\}}{\exp\{\mathbf{z}^\top \tilde{\mathbf{z}} + \sum_i \mathbf{z}^\top \mathbf{x}_i\}}$$

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]

Contrastive learning

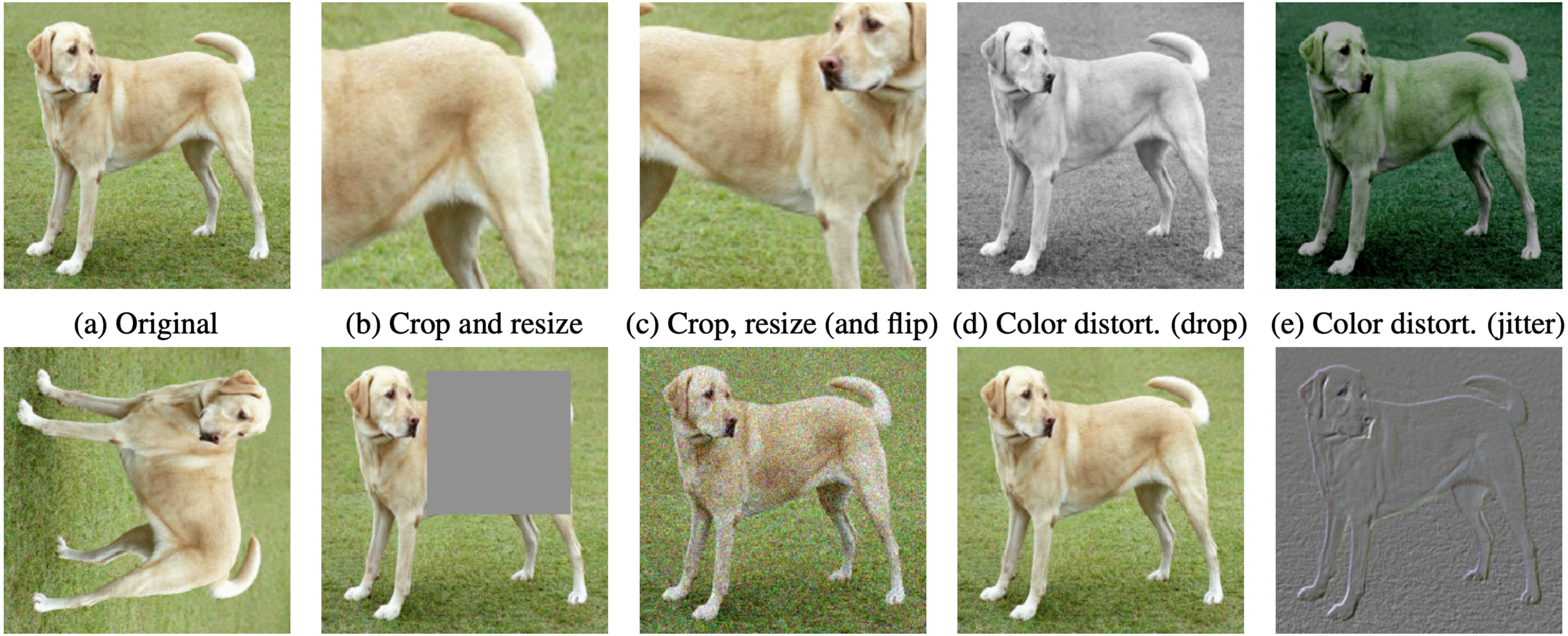


Can build invariance.
Compare to *warped* images.

$$\frac{\exp\{\mathbf{z}^\top \tilde{\mathbf{z}}\}}{\exp\{\mathbf{z}^\top \tilde{\mathbf{z}} + \sum_i \mathbf{z}^\top \mathbf{x}_i\}}$$

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]

Contrastive learning

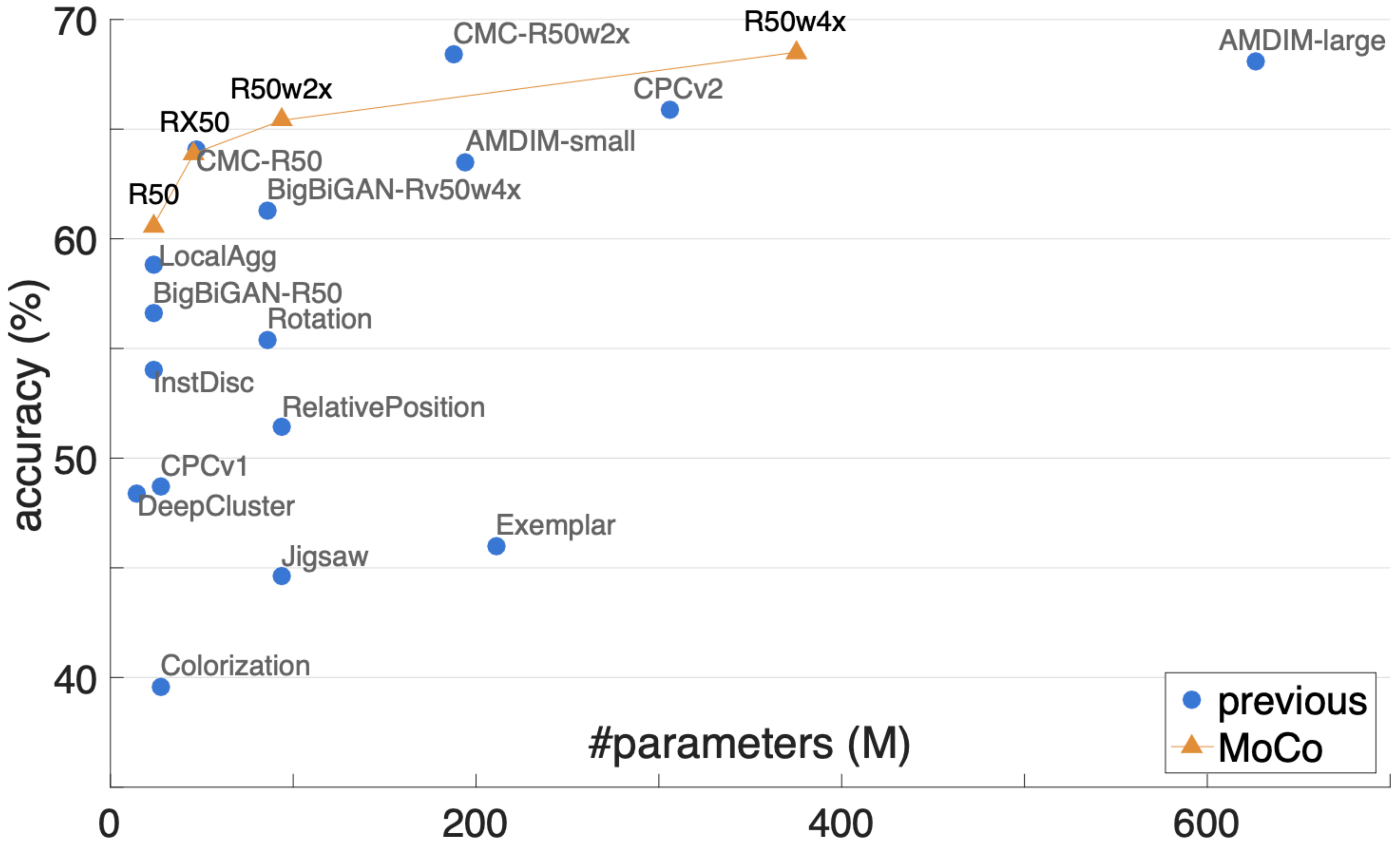


(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

From [Chen et al., SinCLR, 2020]

Performance snapshot



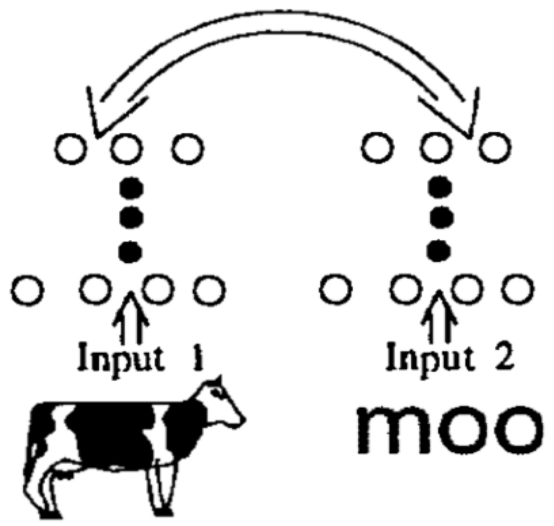
pre-train	AP_{50}
random init.	52.5
super. IN-1M	80.8
MoCo IN-1M	81.4 **(+0.6)**
MoCo IG-1B	82.1 **(+1.3)**

ImageNet linear classification

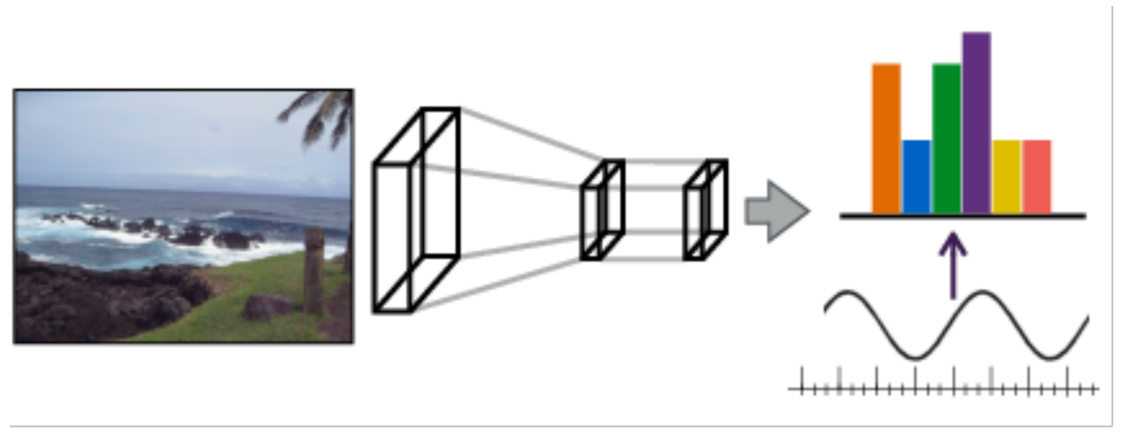
Object detection finetuning

Comparable in many cases to supervised pretraining!

Audio

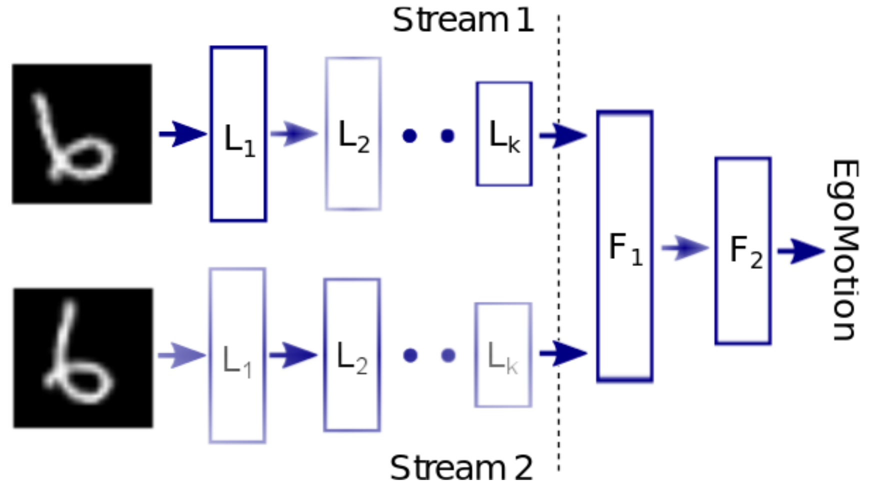


de Sa. NIPS 1994.



Owens et al. ECCV 2016.

Egomotion

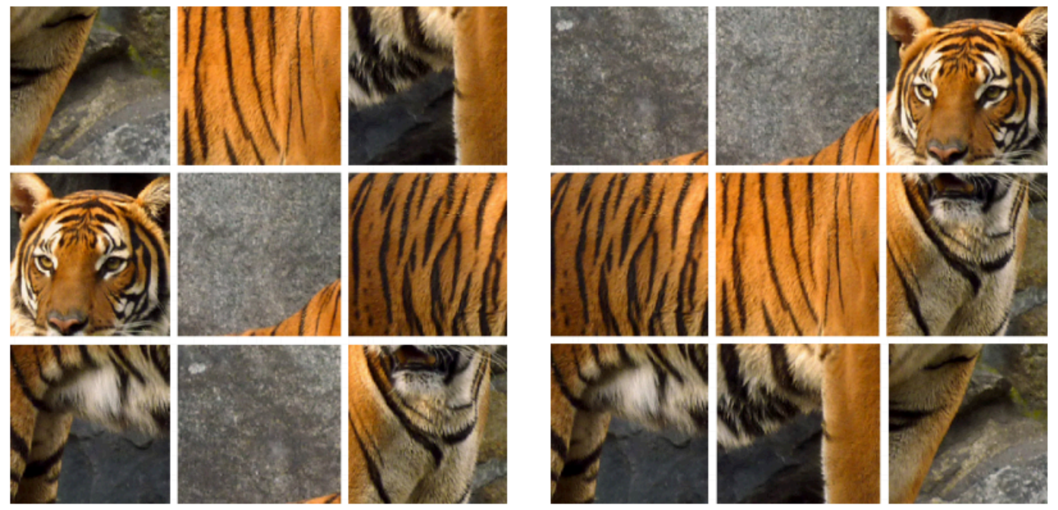


Agrawal et al. ICCV 2015.

Jayaraman et al. ICCV 2015.

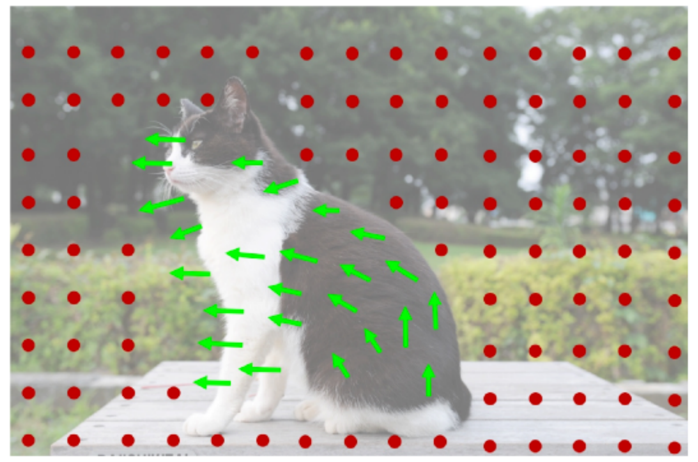
Context

Pathak et al. CVPR 2016.



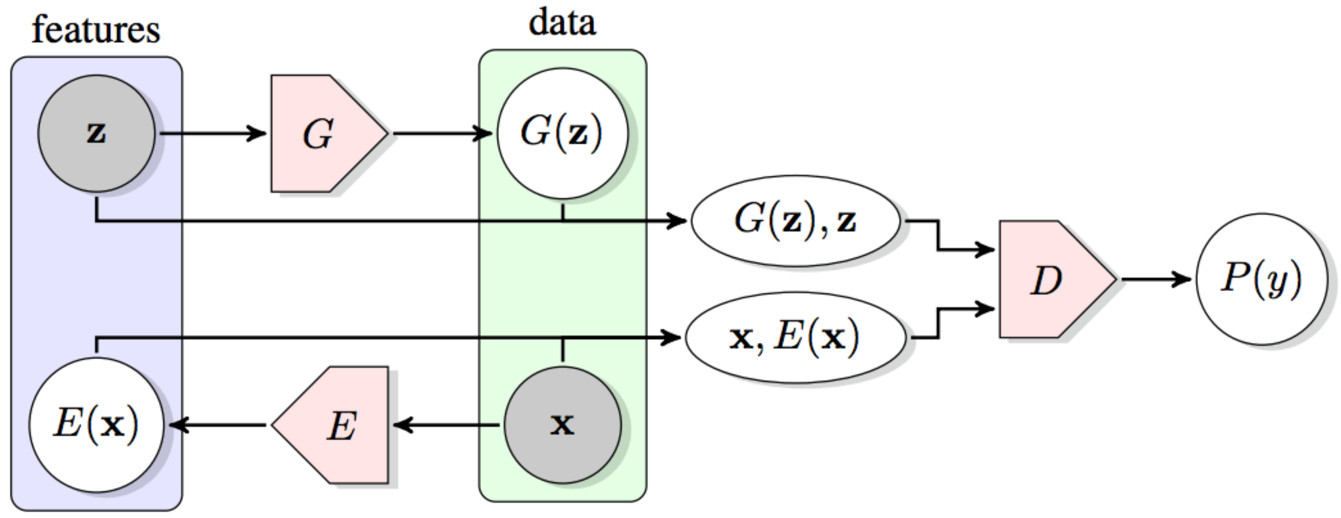
Noroozi and Favaro. ECCV 2016.
Doersch et al. ICCV 2015.

Video



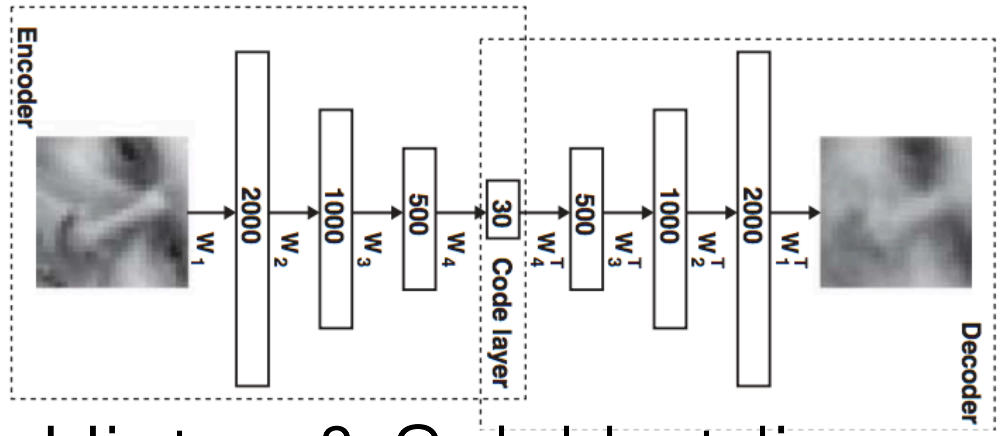
Wang et al. ICCV 2015. Pathak et al. CVPR 2017.
Misra et al. ECCV 2016.

Generative Modeling



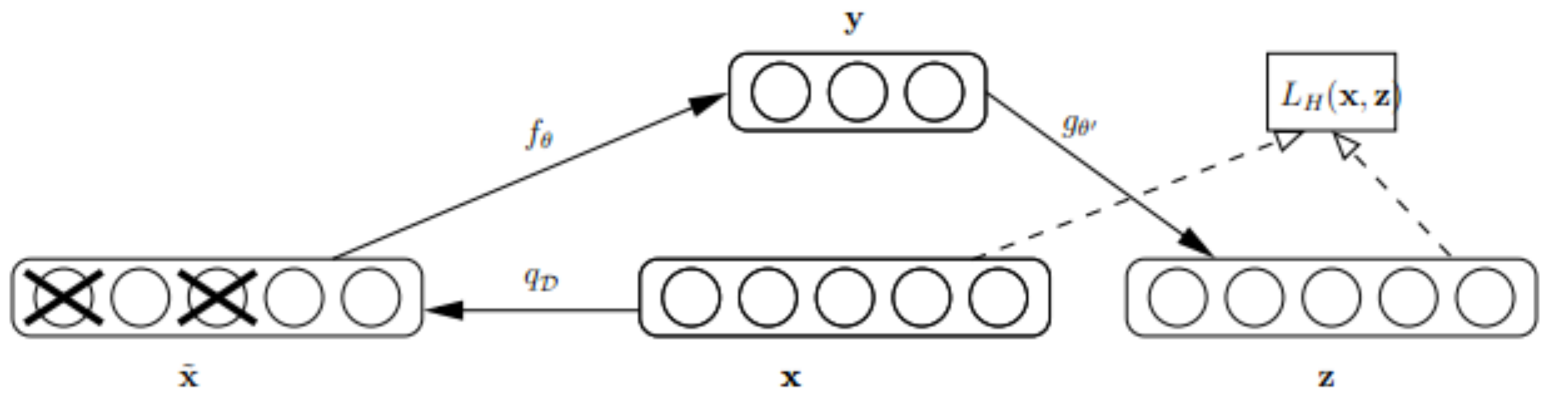
Donahue et al. Dumoulin et al. ICLR 2017.

Autoencoders



Hinton & Salakhutdinov.
Science 2006.

Denoising Autoencoders



Vincent et al. ICML 2008.

Goal: Set up a pre-training scheme to induce a “useful” representation

[Slide credit: Richard Zhang]

Summary

1. Deep nets learn *representations*
2. This is useful because representations transfer — they act as prior knowledge that enables quick learning on new tasks
3. Representations can also be learned without labels
4. Without labels there are many ways to learn representations. We saw:
    1. representations as compressed codes
    2. representations that are predictive of missing data

Source: Isola, Freeman, Torralba

Next time: Sight, sound, and touch