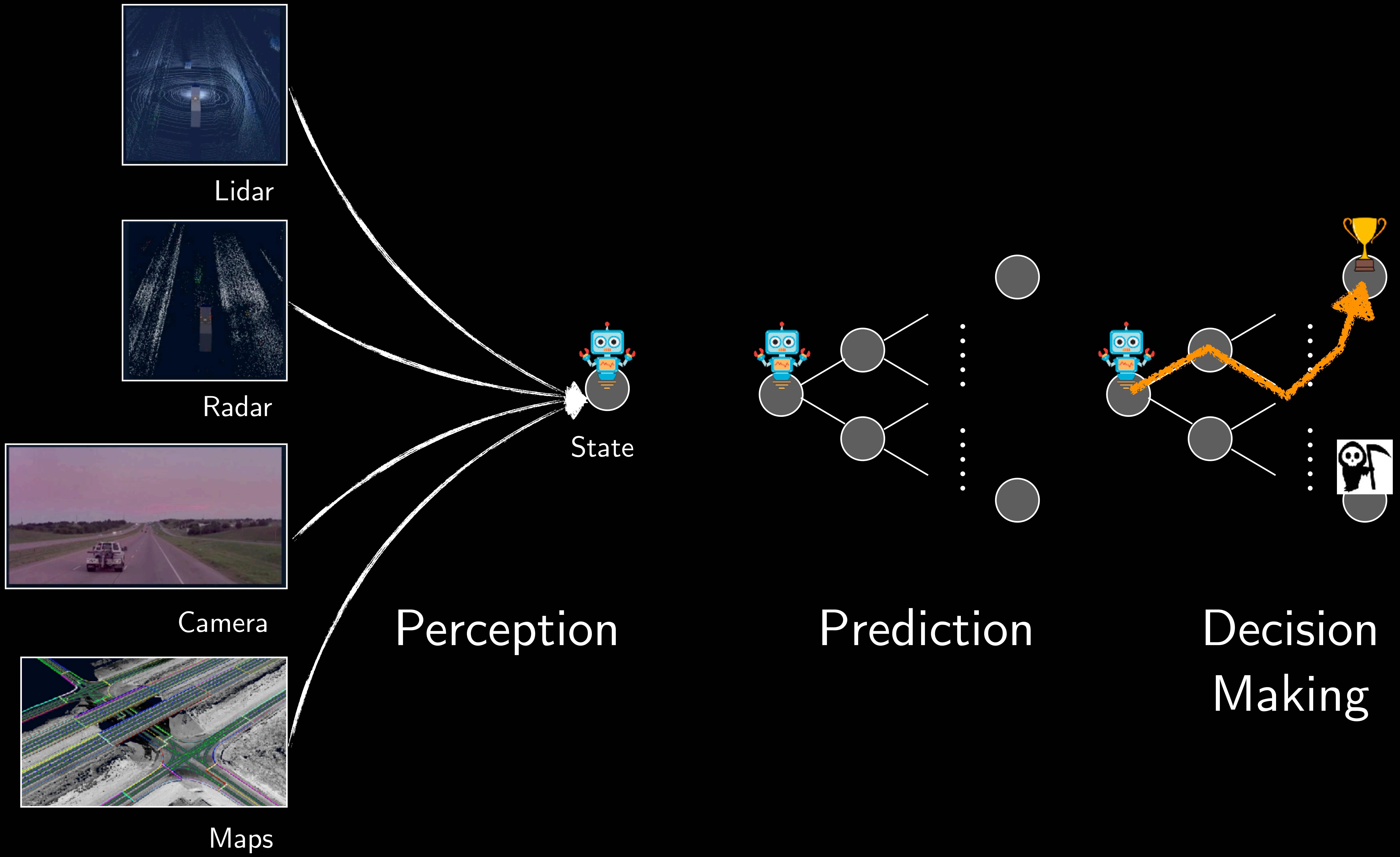
Visual Representation Learning

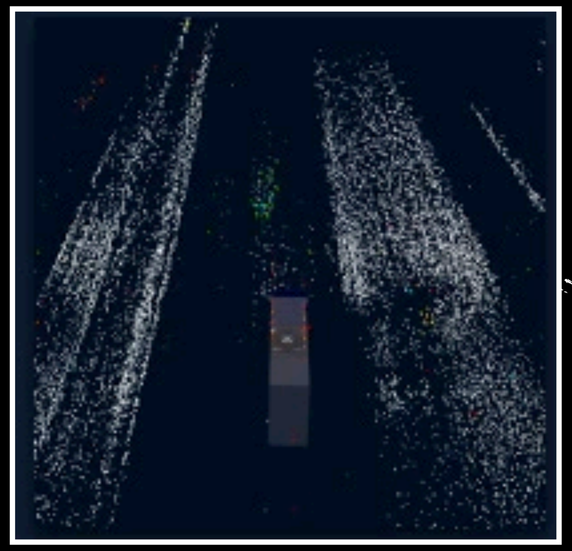
Sanjiban Choudhury

Cornell Bowers CIS
Computer Science


Lidar
Radar
Camera
Maps
State
Perception
Prediction
Decision Making

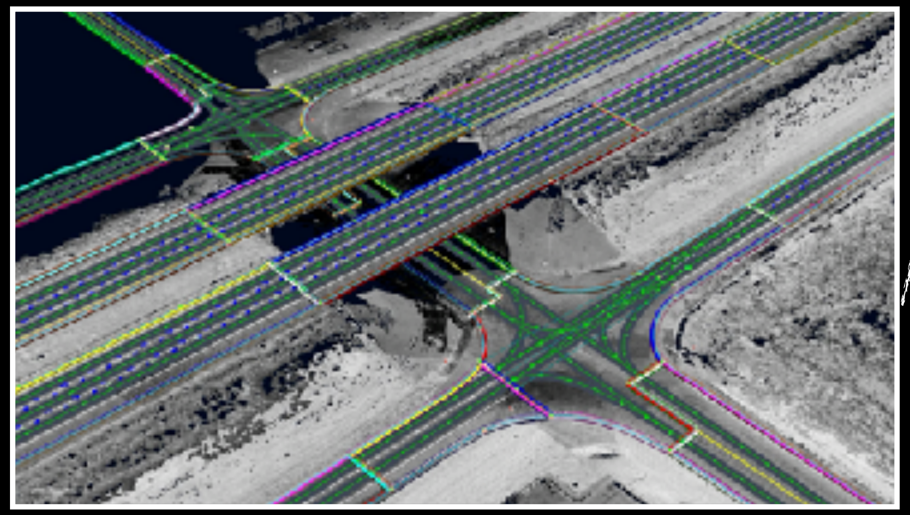
Past classes ...

Lidar



Radar

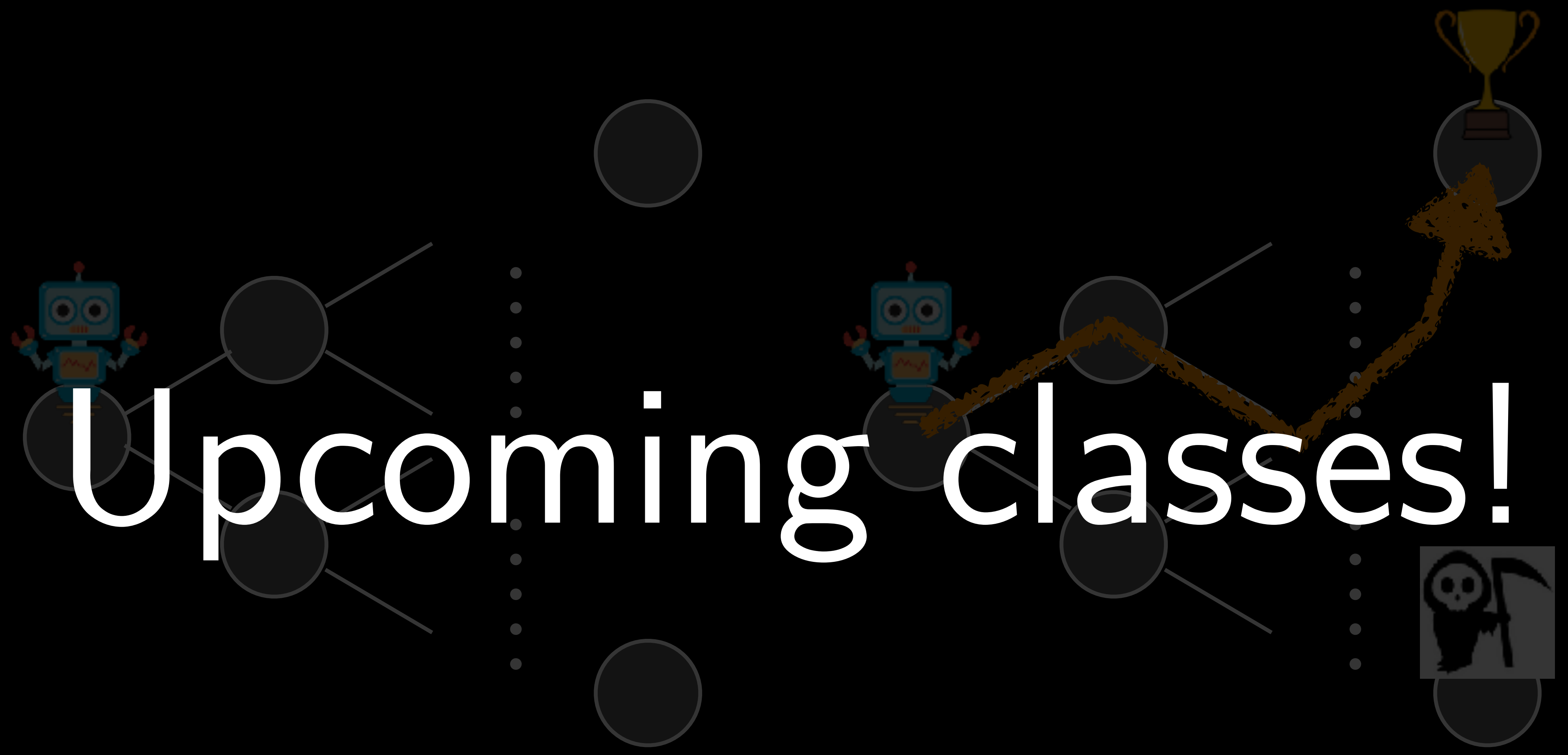
Camera



Maps

State

Perception



Upcoming classes!

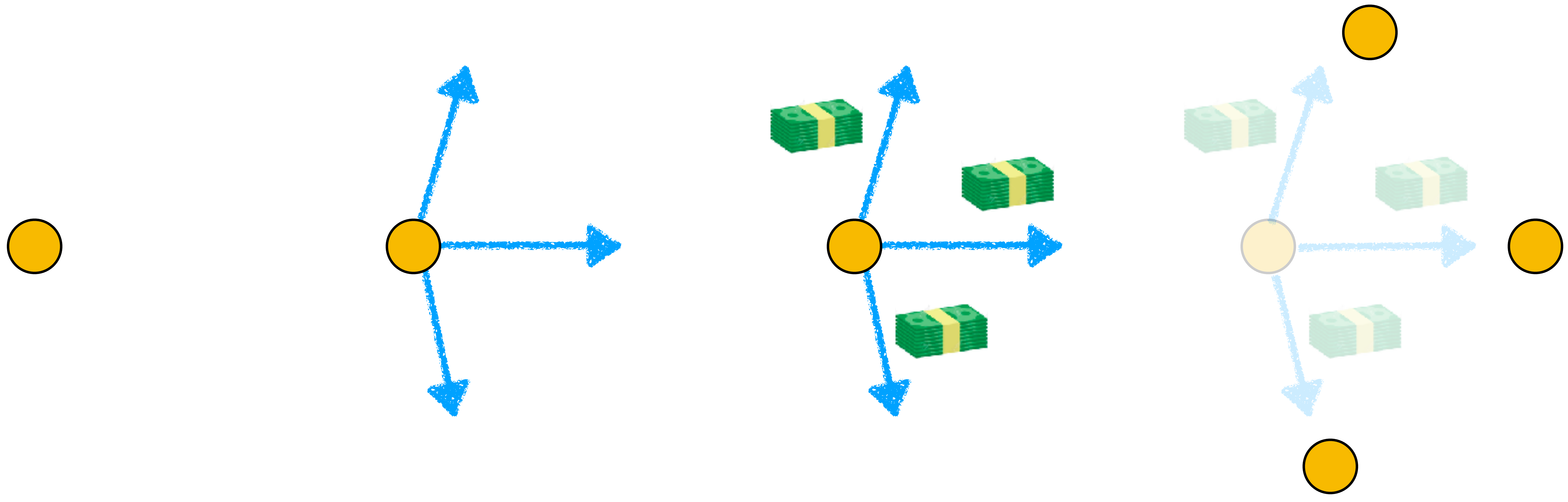
Prediction

Decision Making

Markov Decision Process

A mathematical framework for modeling sequential decision making

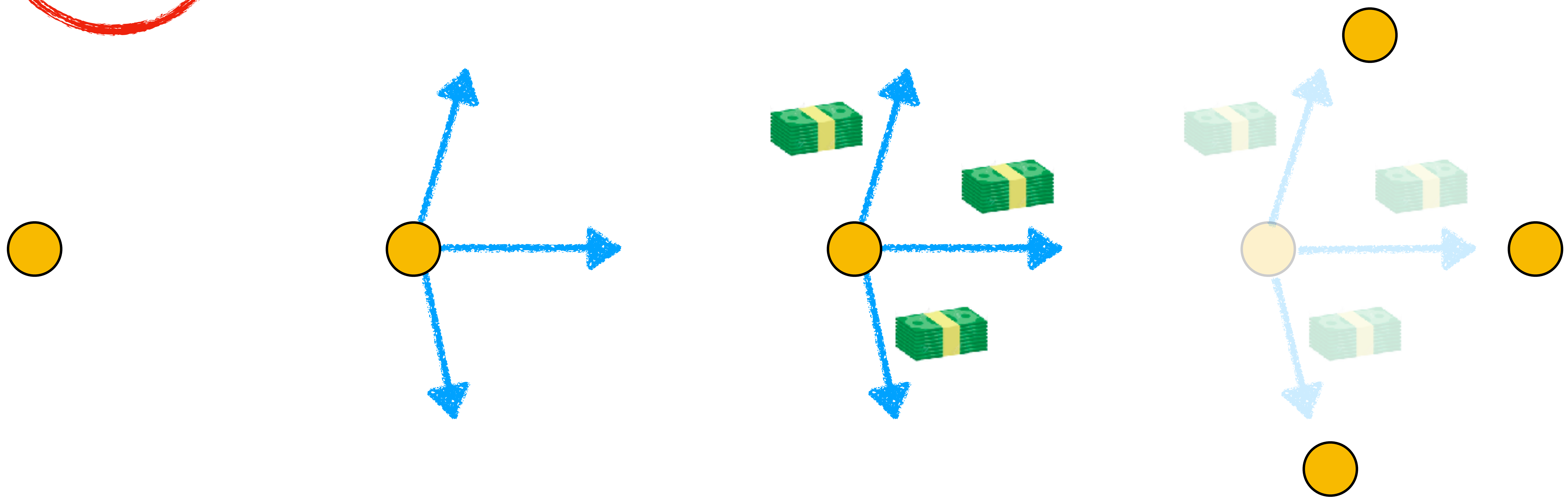
$$< S \, , A \, , C \, , \mathcal{T} >$$



How do we estimate state?

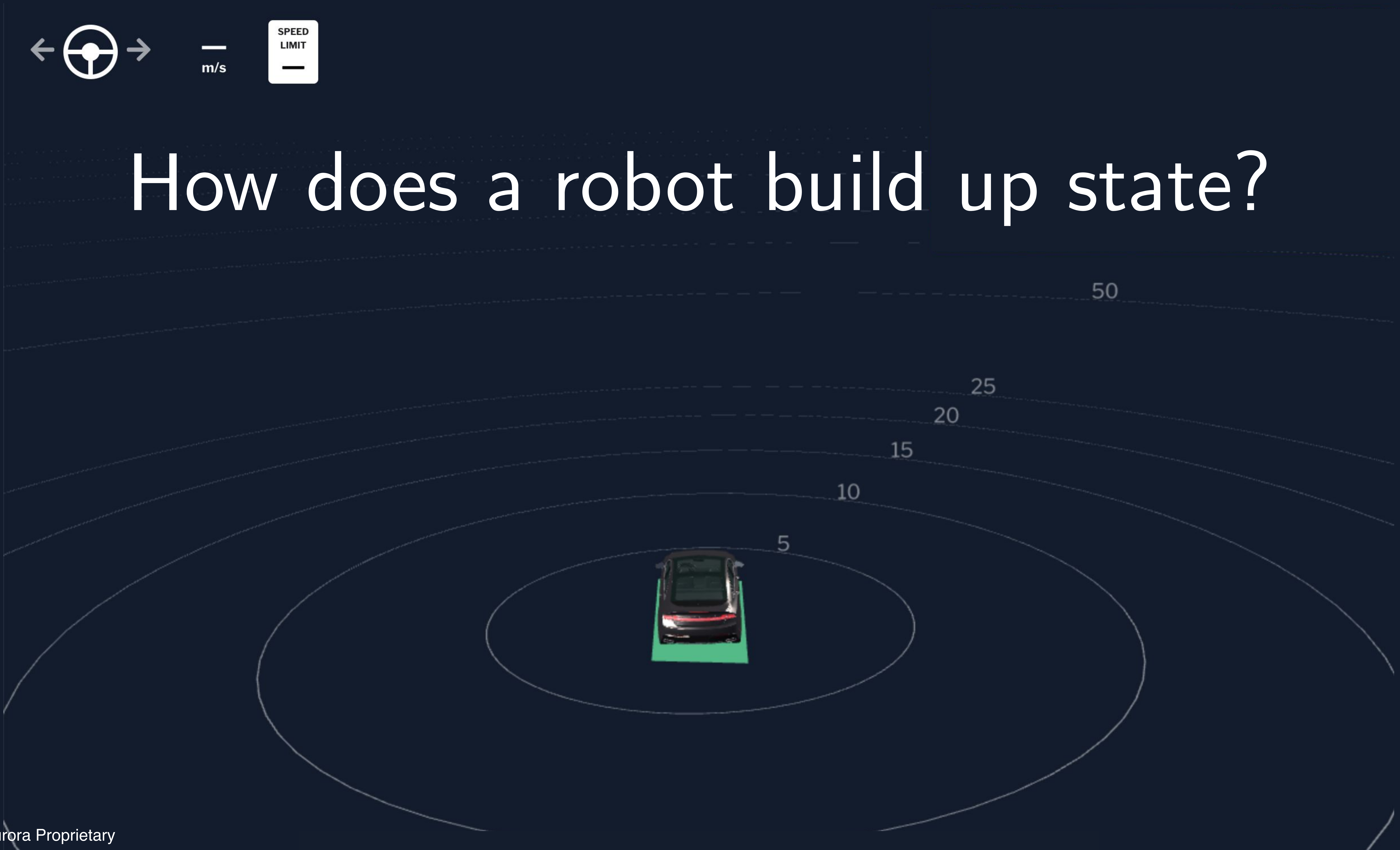
A mathematical framework for modeling sequential decision making

$$< S , A , C , \mathcal{T} >$$



Self-driving

How does a robot build up state?




0.0
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10

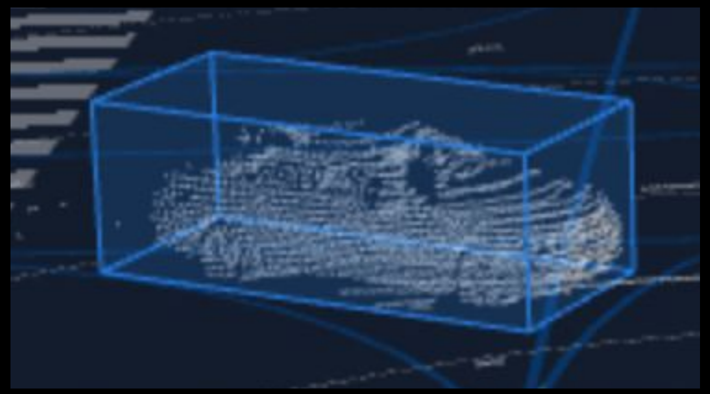
0.0
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10
5


0.0
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10


1.2
m/s
SPEED
LIMIT
11
ACTIVE
50
25
20
15
10

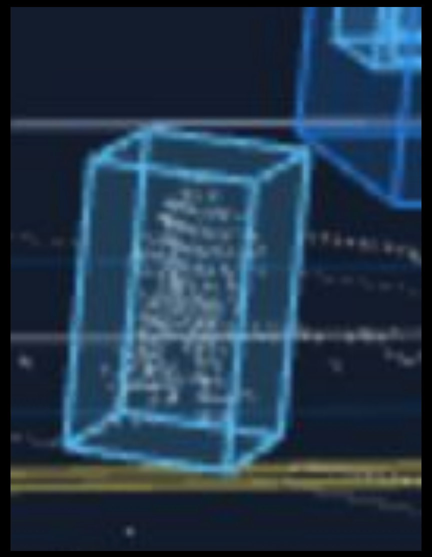
pose

(x, y, ψ)



vel

$(\dot{x}, \dot{y}, \dot{\psi})$



type

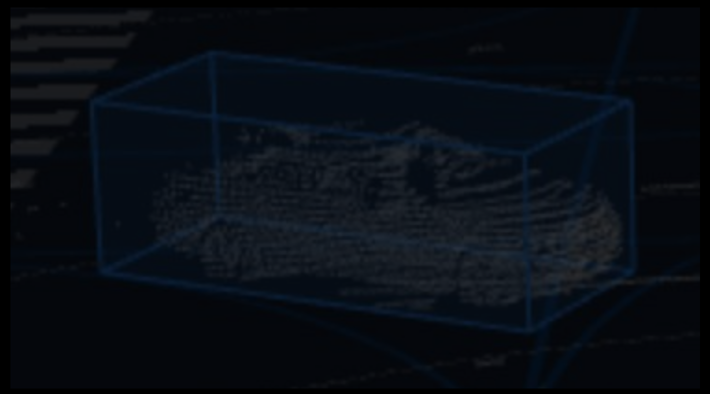
(pedestrian, car, cyclist)

But we do not observe these directly!

$$s_t \xrightarrow{a_t} s_{t+1} \xrightarrow{a_{t+1}} s_{t+2}$$

pose

(x, y, ψ)



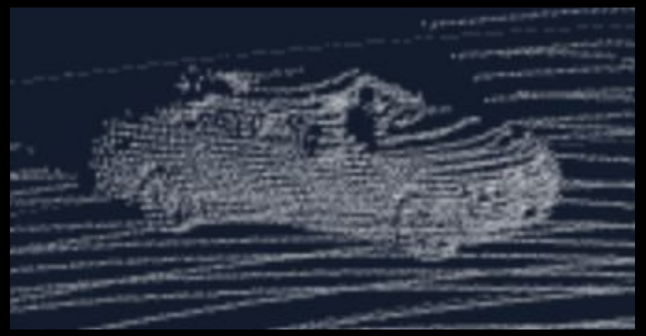
vel

$(\dot{x}, \dot{y}, \dot{\psi})$

type

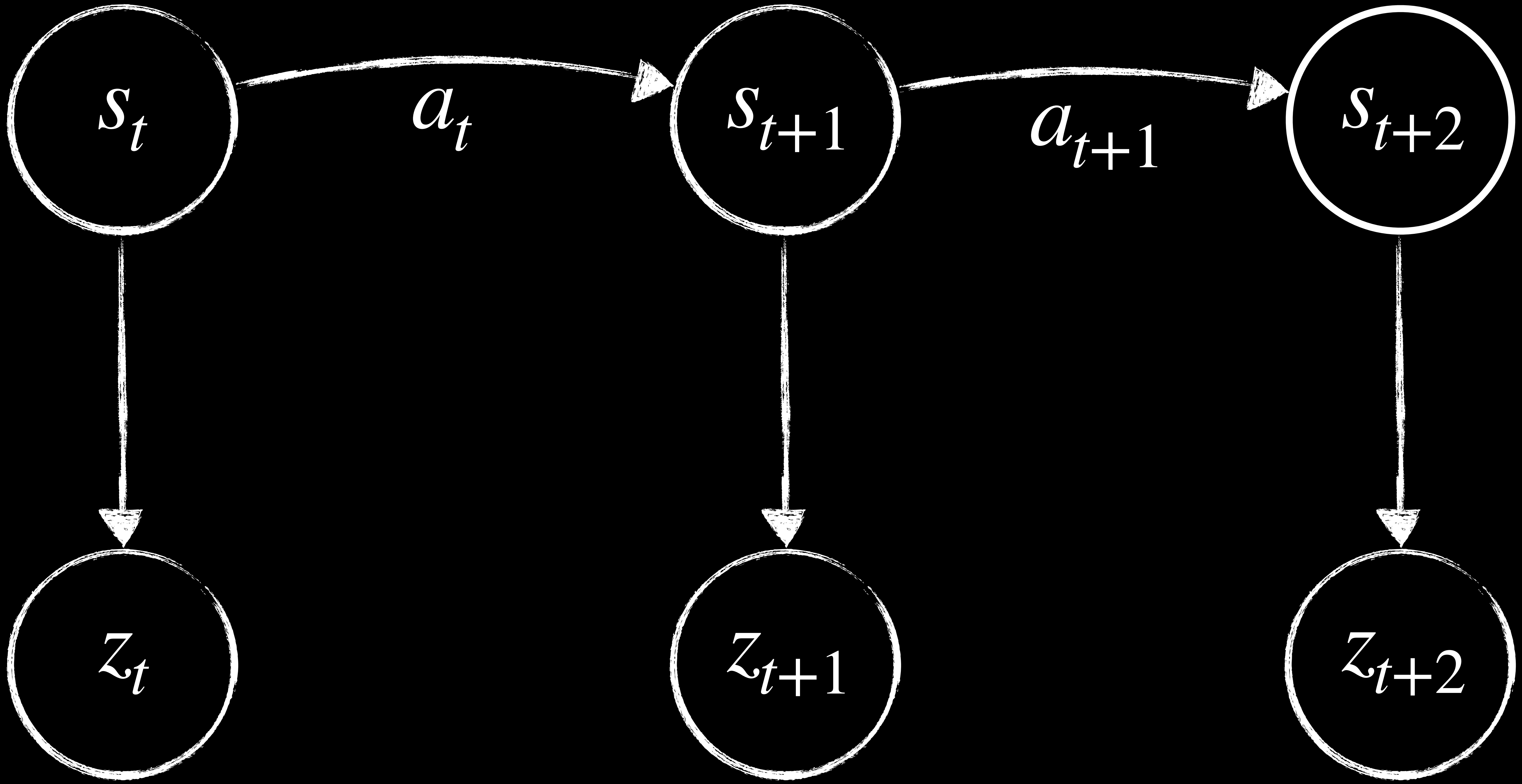
(pedestrian, car, cyclist)

camera



lidar

Estimate state from observations



Learning a state representation

z_t s_t

Goal: Learn a state representation

$$P(s_t \mid z_t, z_{t-1}, z_{t-2}, \ldots)$$

Another example: Manipulation

$$P(s_t \,|\, z_t, z_{t-1}, z_{t-2}, \ldots)$$

From Jeannette Bohg: CS231A Computer Vision: From 3D Reconstruction to Recognition

What is observation z_t ?

What is state s_t ?

Another example: Manipulation

$$P(s_t \mid z_t, z_{t-1}, z_{t-2}, \ldots)$$

What is observation z_t ?

RGB Image + Depth (Kinect)

What is state s_t ?

6D Pose of the object

From Jeannette Bohg: CS231A Computer Vision: From 3D Reconstruction to Recognition

Today's Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by
unsupervised learning and **self-supervised** learning?

Today's Class

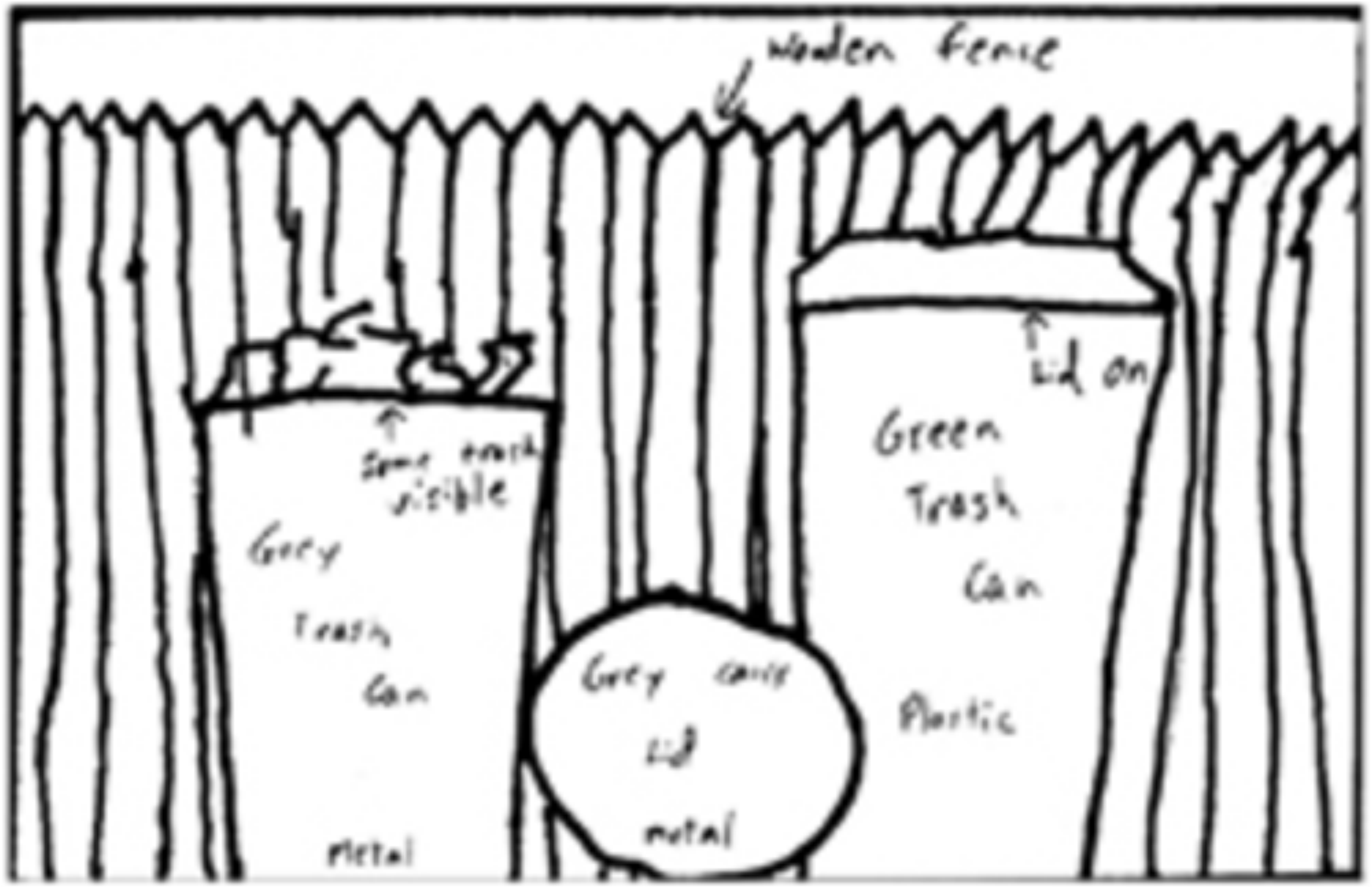
What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by **unsupervised** learning and **self-supervised** learning?

Observed image

Drawn from memory

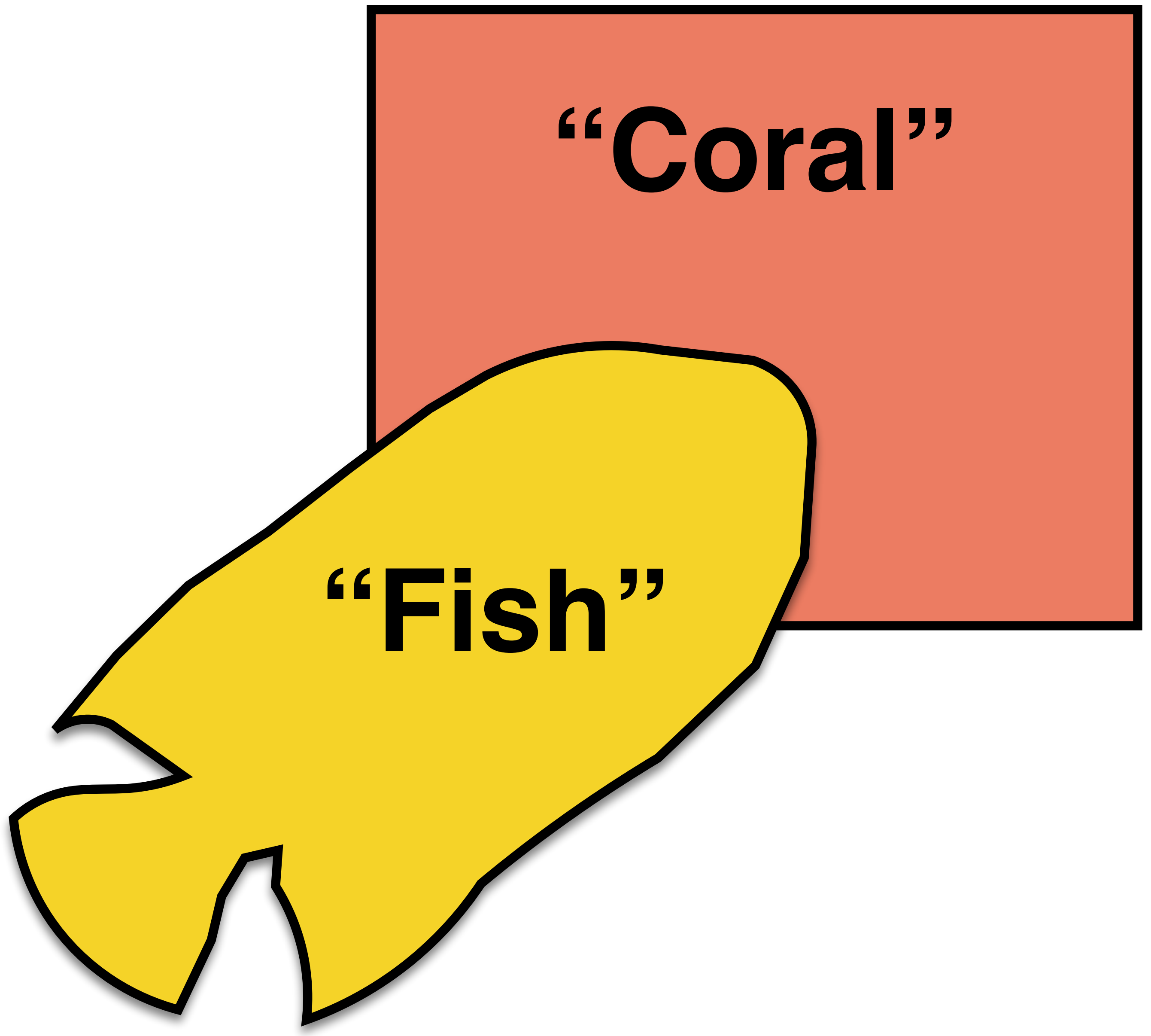


[Bartlett, 1932]
[Intraub & Richardson, 1989]

Representation learning

$\mathbf{x}$

Image



Compact mental representation

From MIT 6.8300/6.8301: Advances in Computer Vision

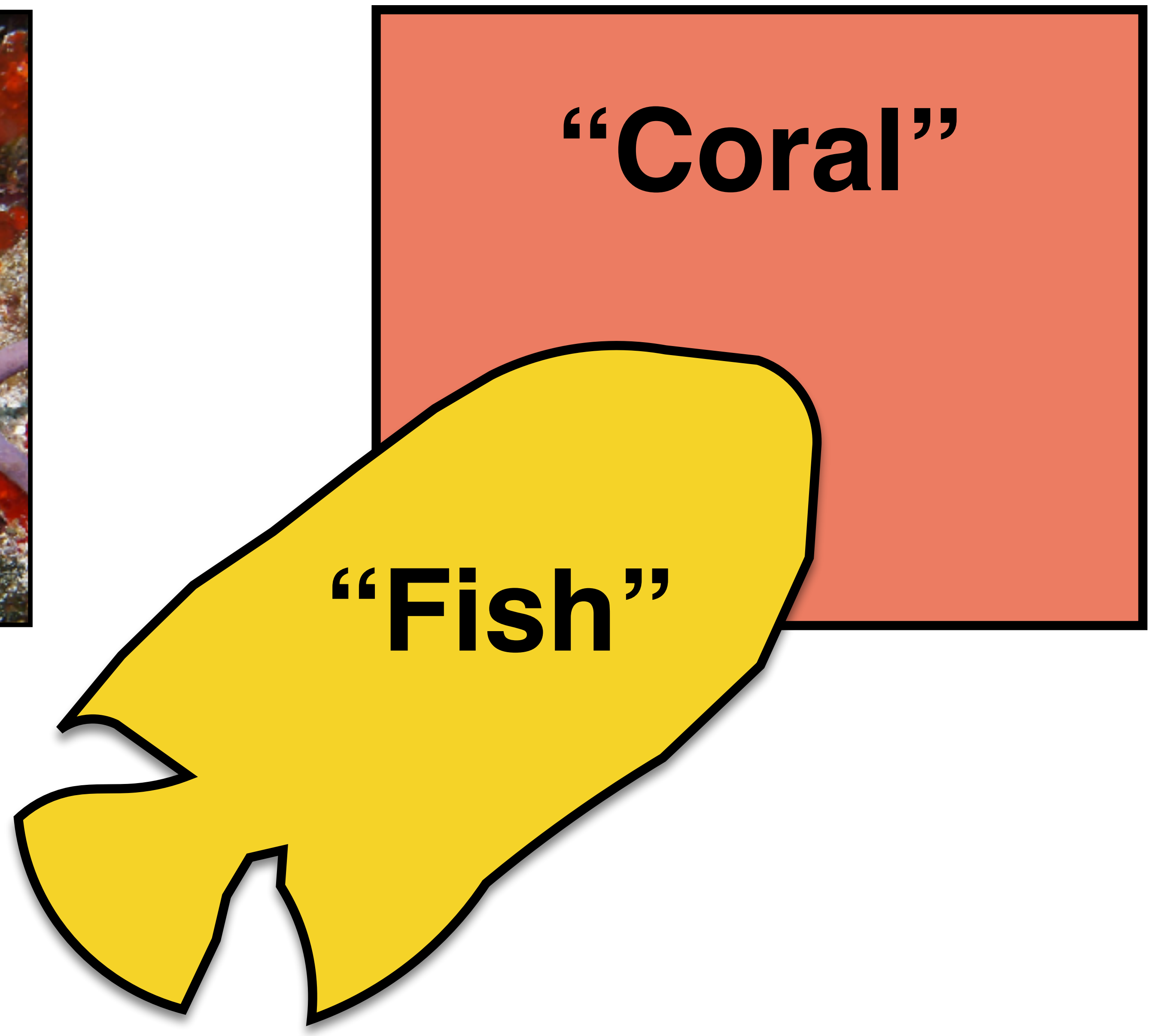
Activity!

Think-Pair-Share!

Think (30 sec): What makes a representation good?

Pair: Find a partner

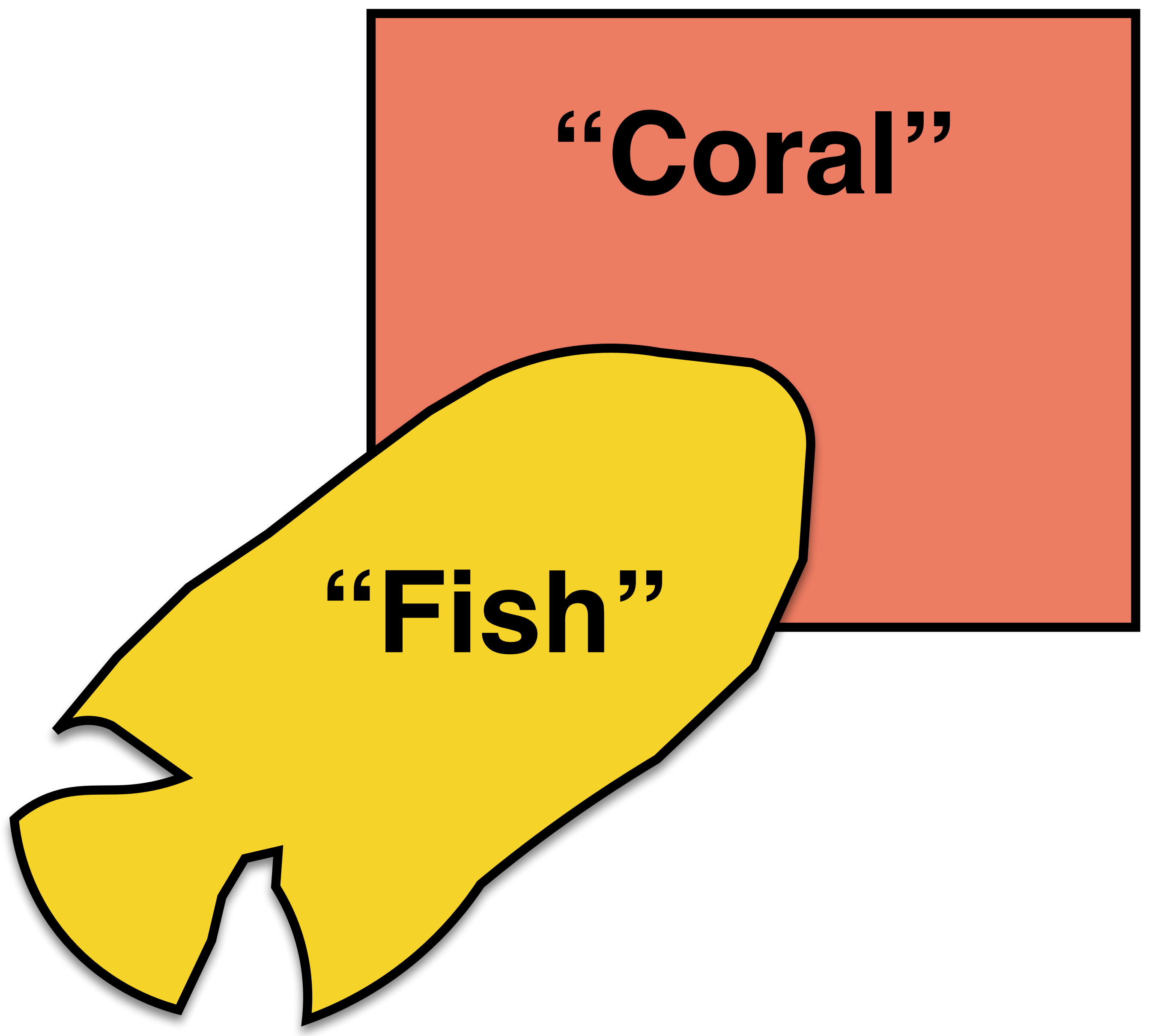
Share (45 sec): Partners exchange ideas



Representation learning

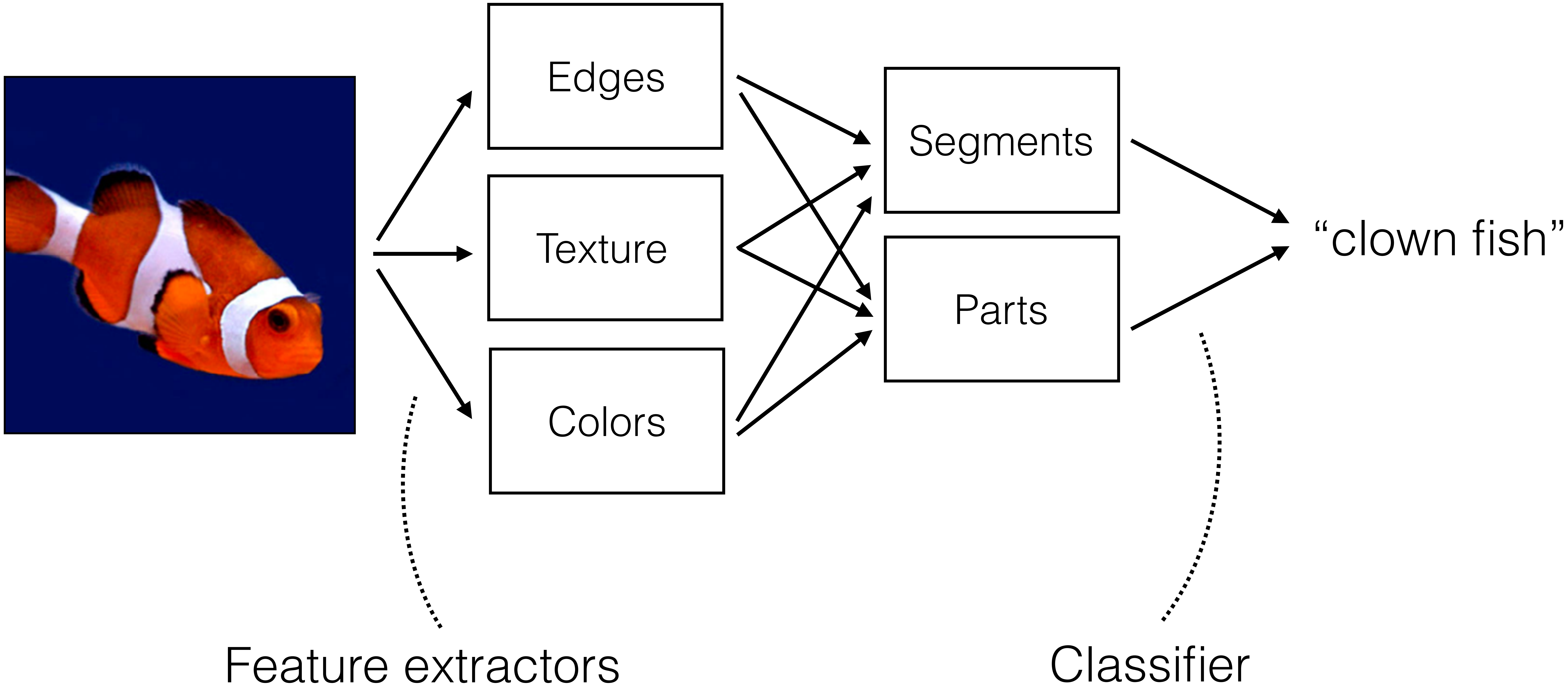
Good representations are:

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



[See “Representation Learning”, Bengio 2013, for more commentary]

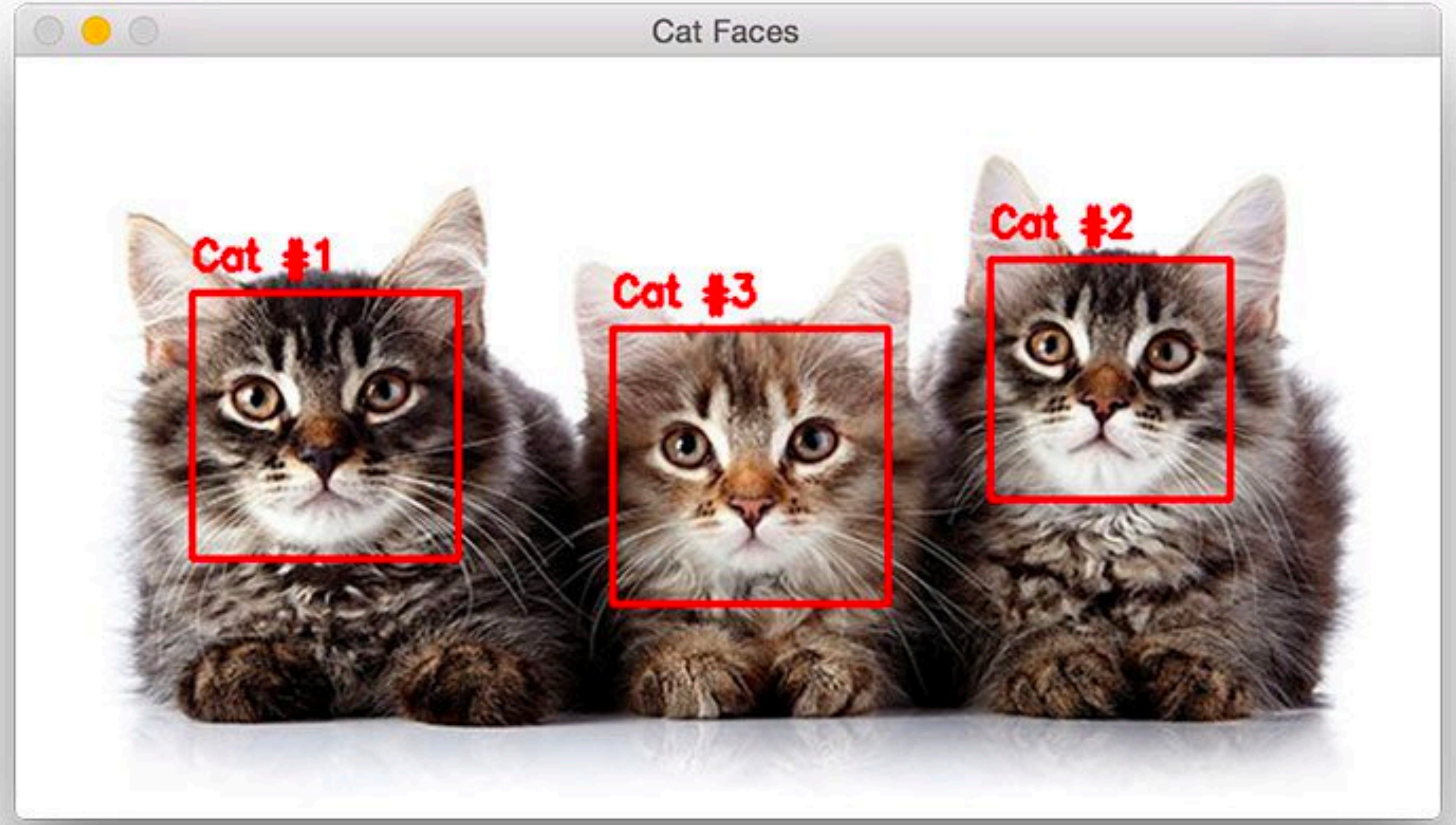
Traditional CV Pipeline



From MIT 6.8300/6.8301: Advances in Computer Vision

Can there be more than one representation that is useful?

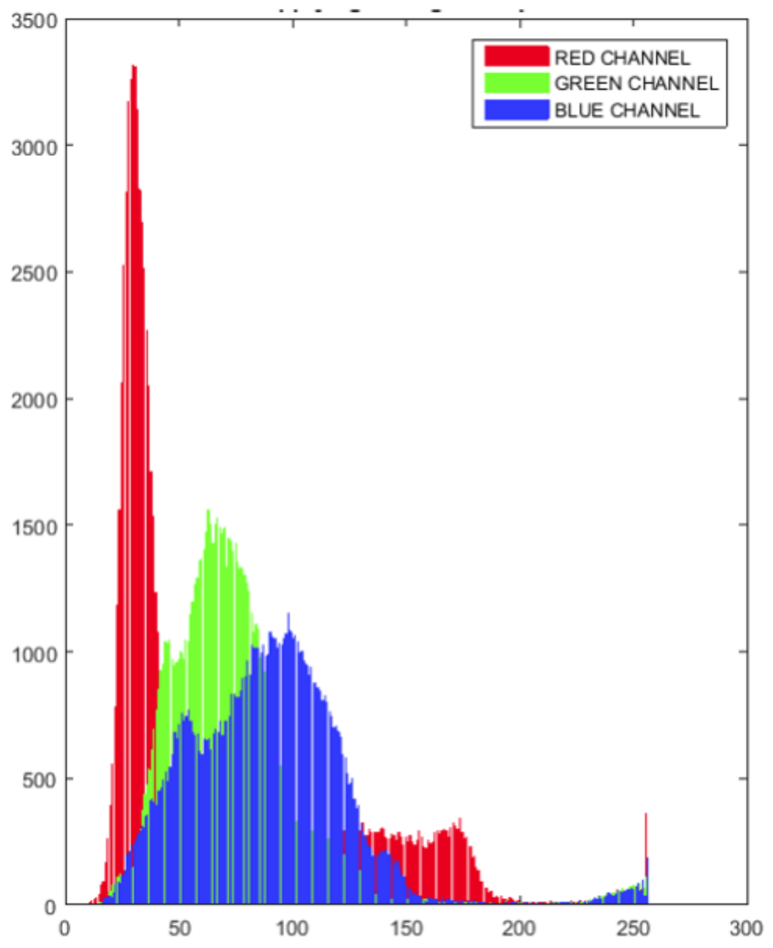
Exercise: Let's think about cats!



Represent these cats with a cat detector





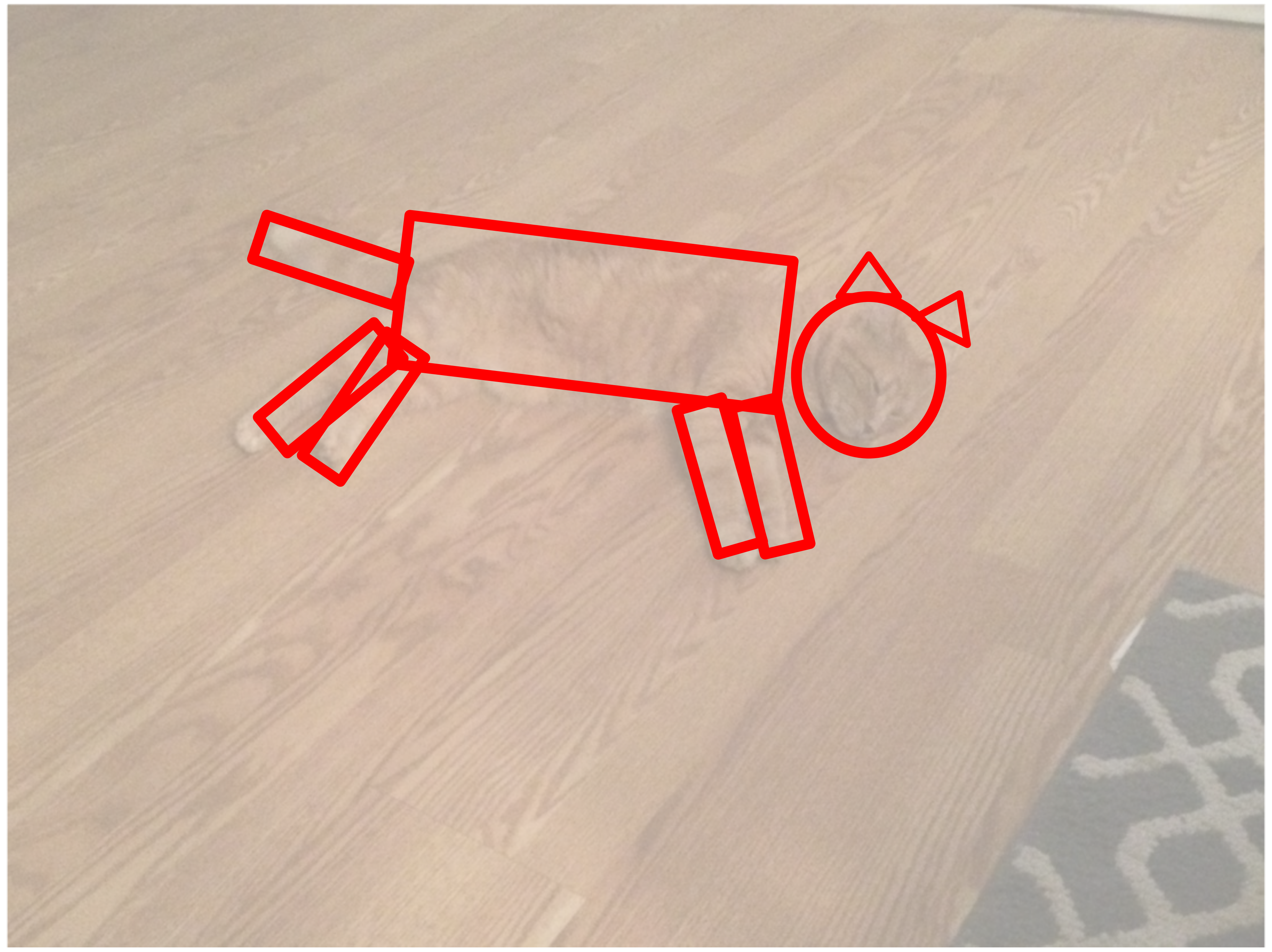
Example from CS331B: Representation Learning in Computer Vision

Represent these cats with a cat detector (II)



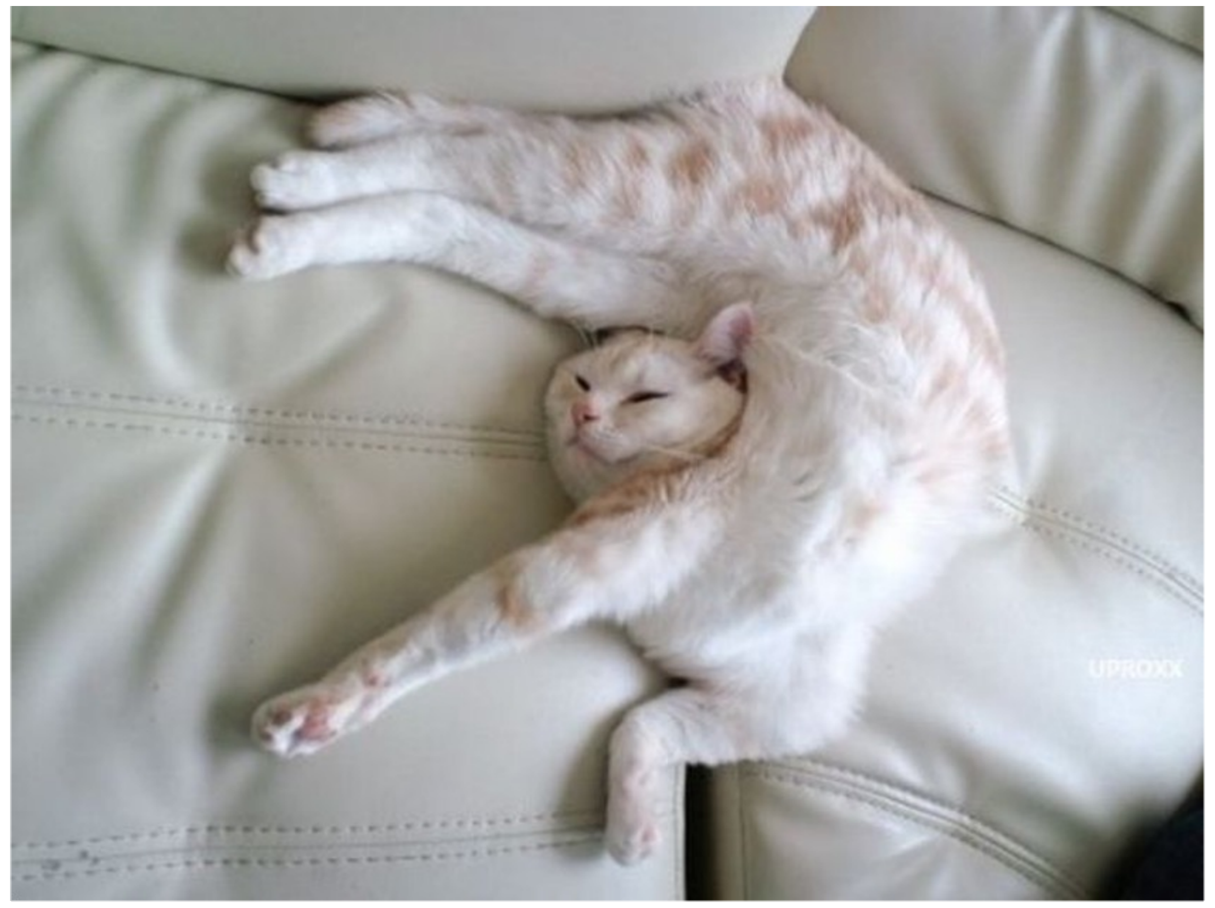
Example from CS331B: Representation Learning in Computer Vision

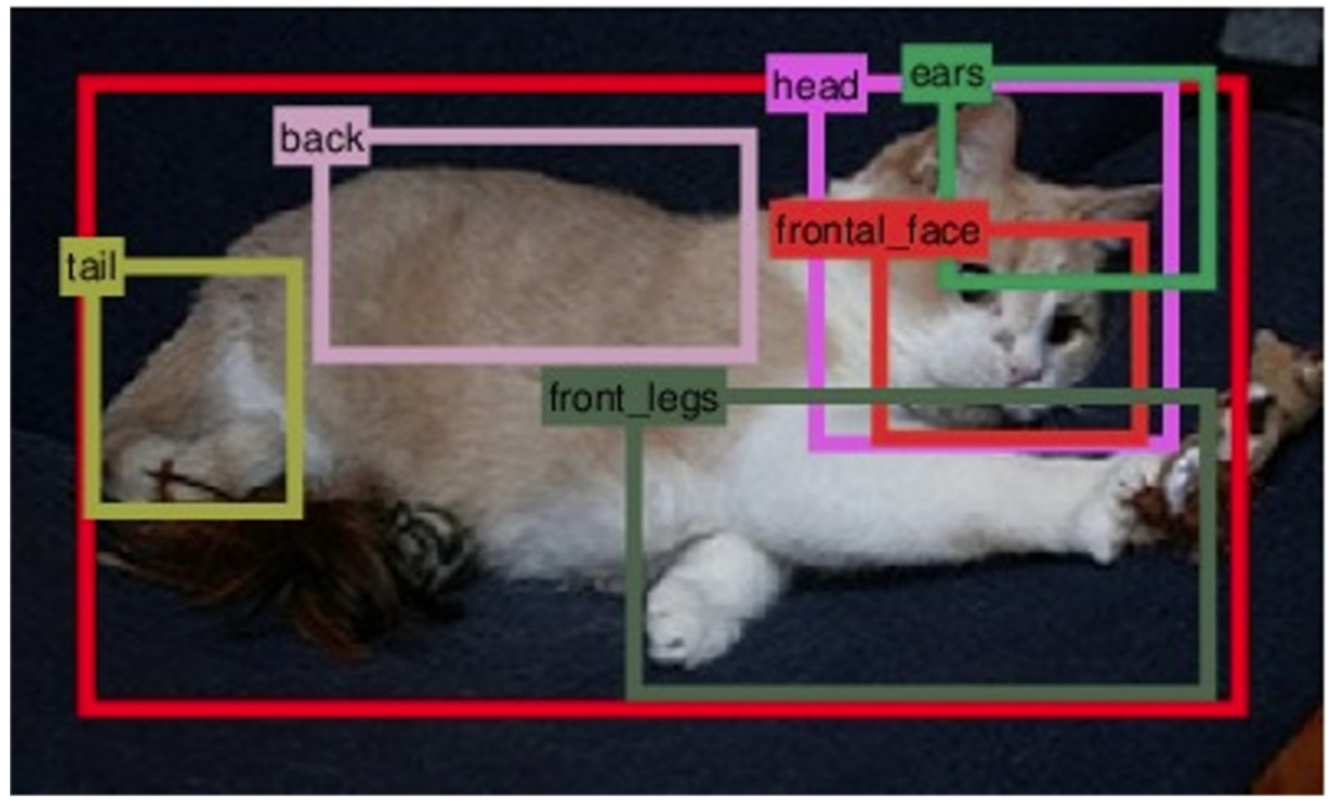
Represent these cats with a cat detector (II)



Example from CS331B: Representation Learning in Computer Vision

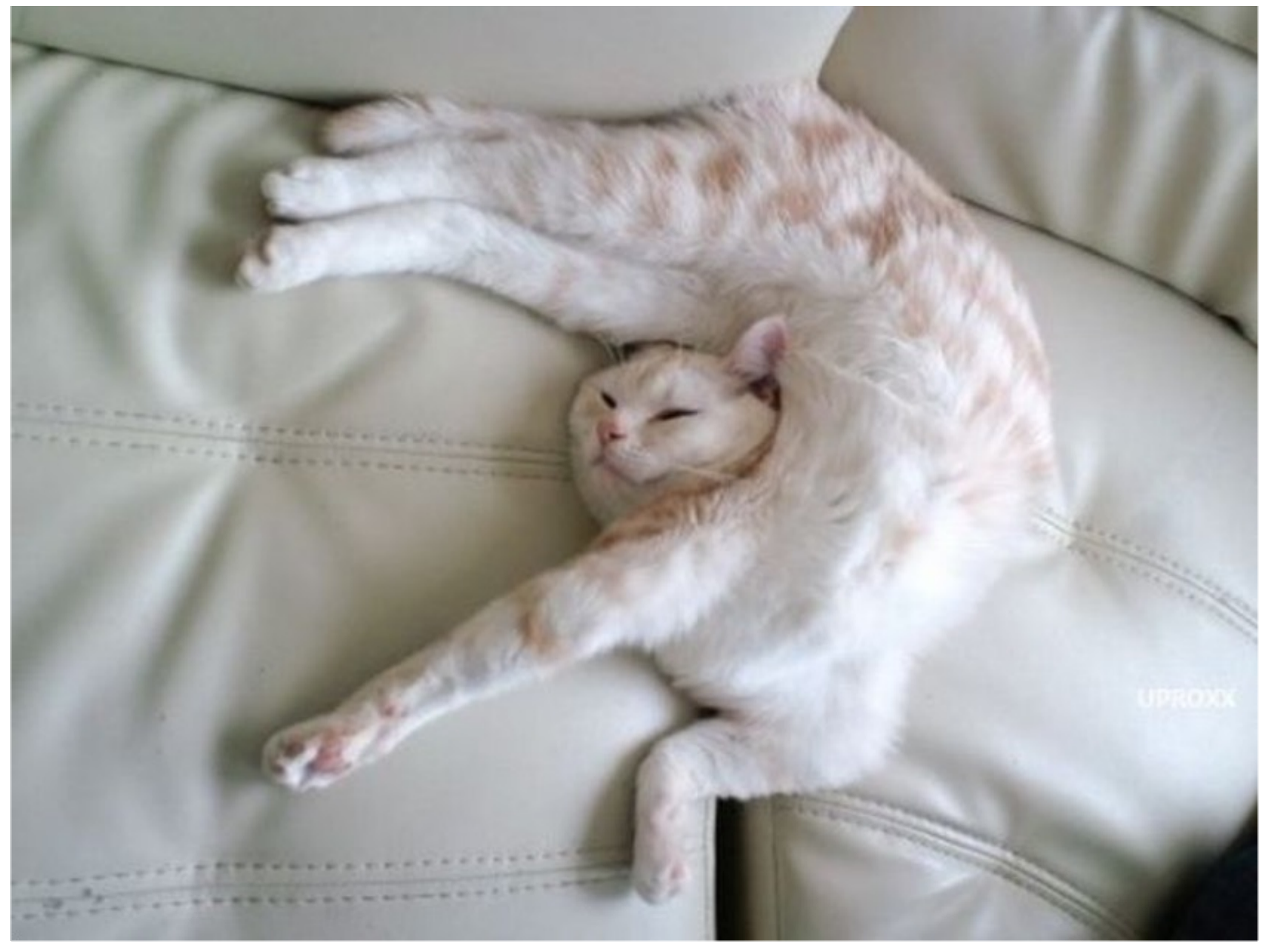
Represent these cats with a cat detector (III)

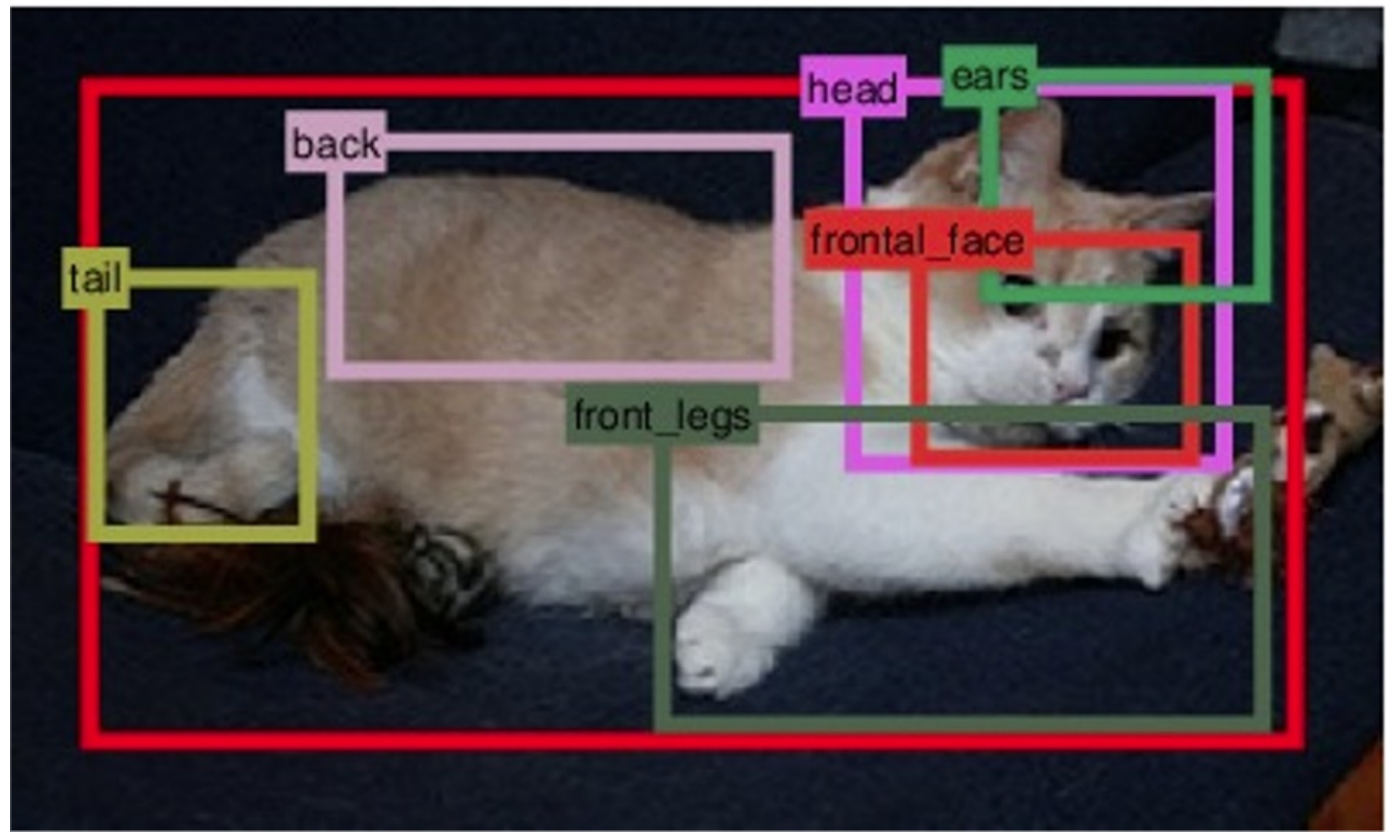




Example from CS331B: Representation Learning in Computer Vision

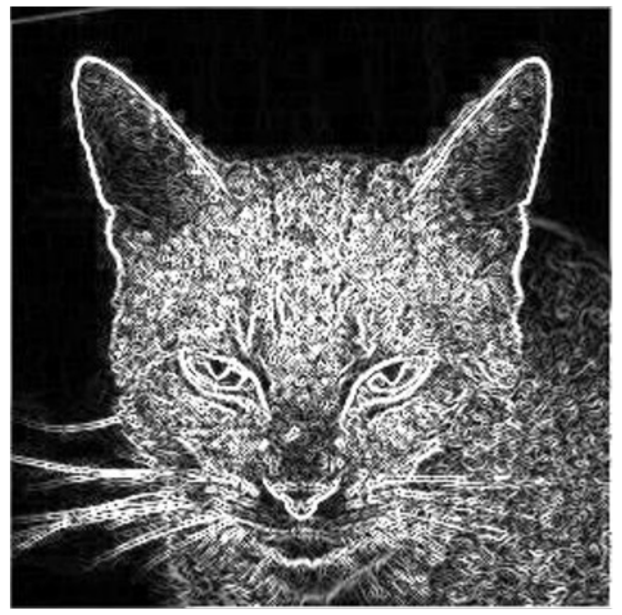
Represent these cats with a cat detector (III)

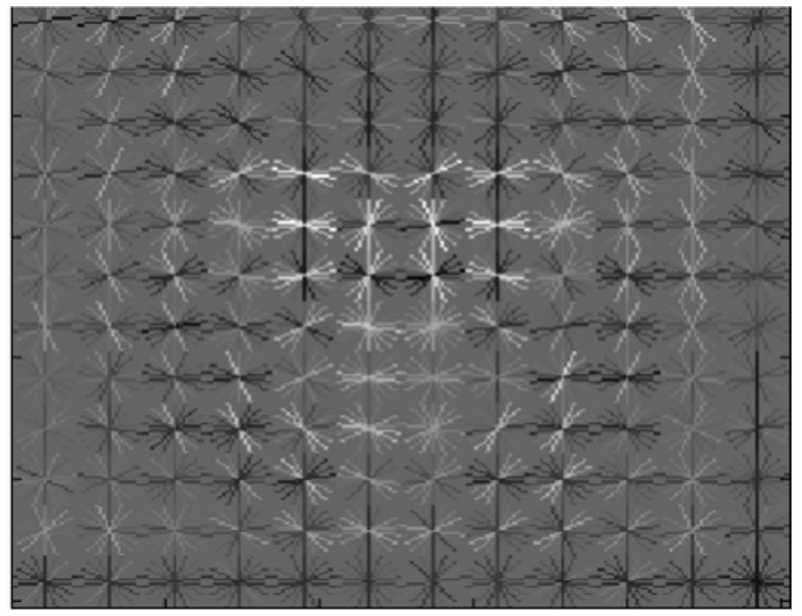




Example from CS331B: Representation Learning in Computer Vision

Represent these cats with a cat detector (IV)

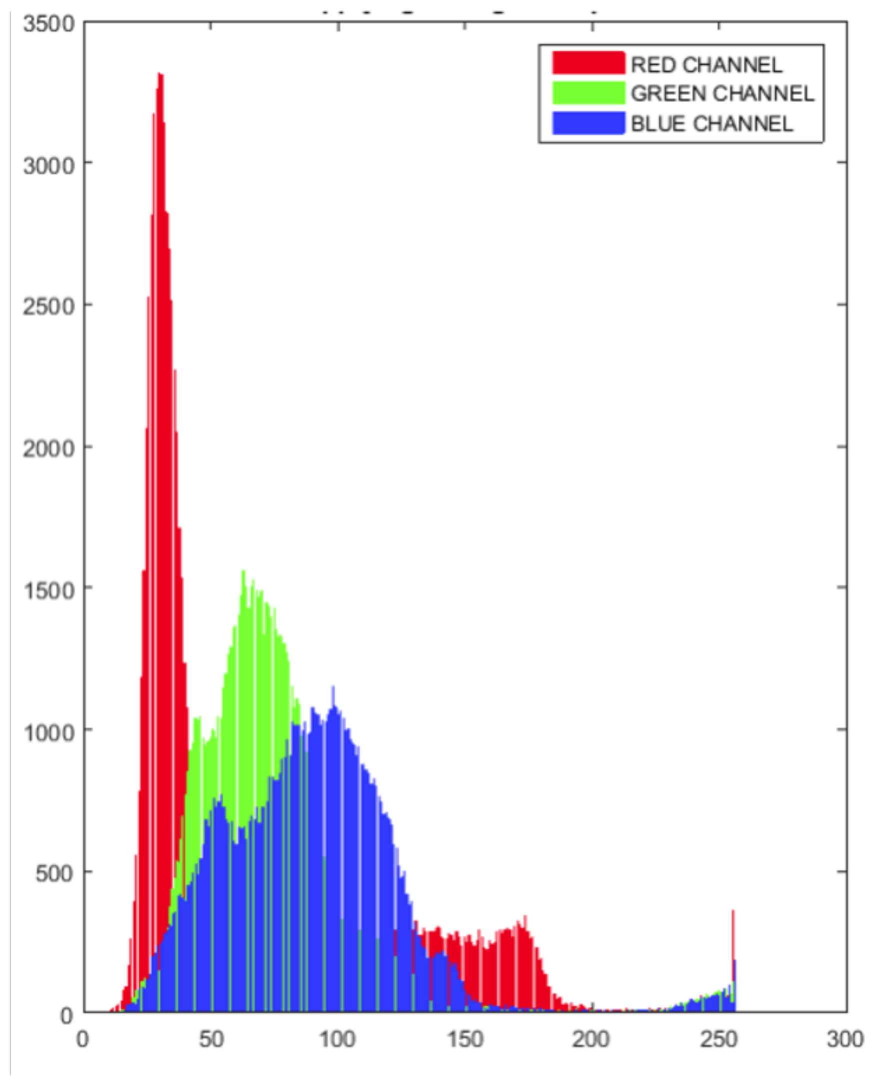


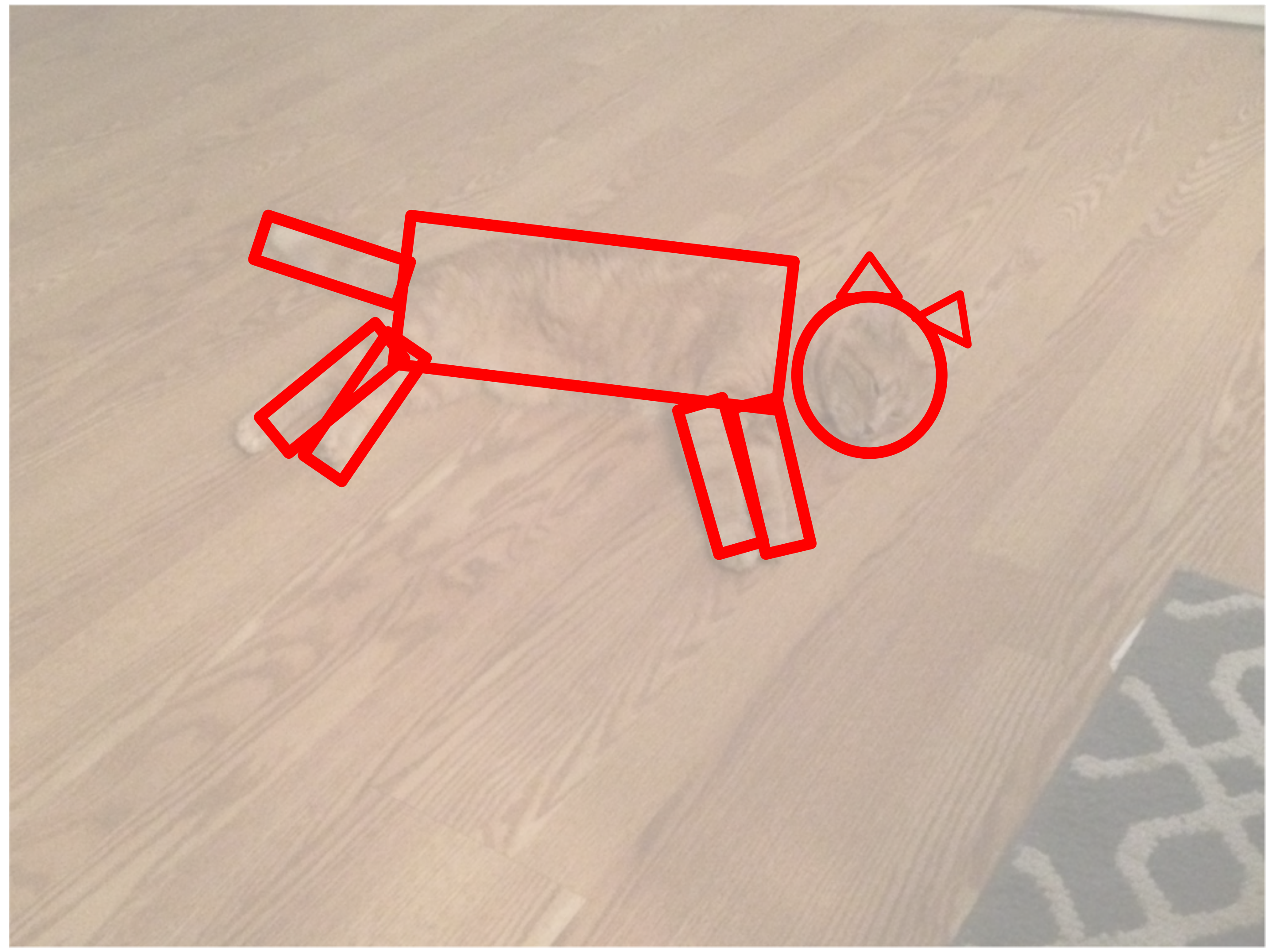


Example from CS331B: Representation Learning in Computer Vision

Summary of representations

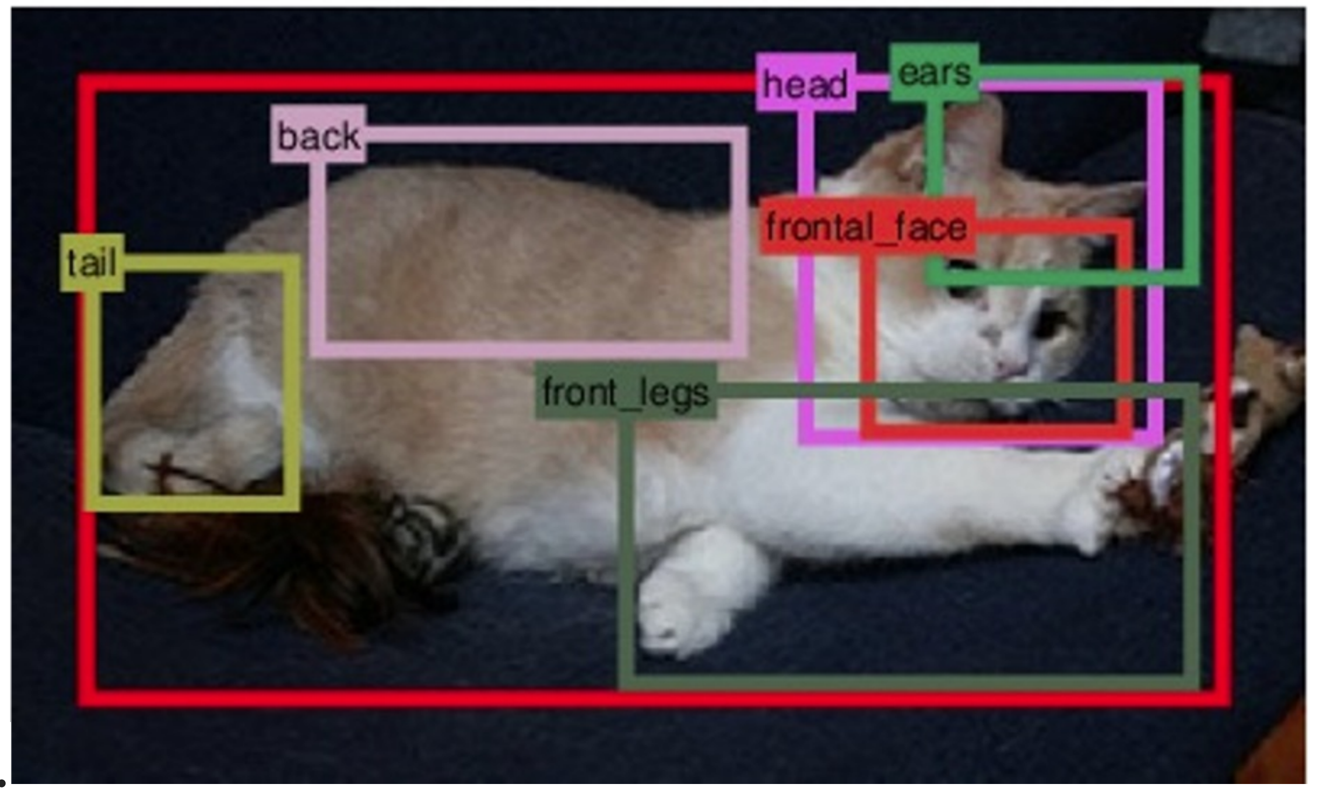
Color Histograms

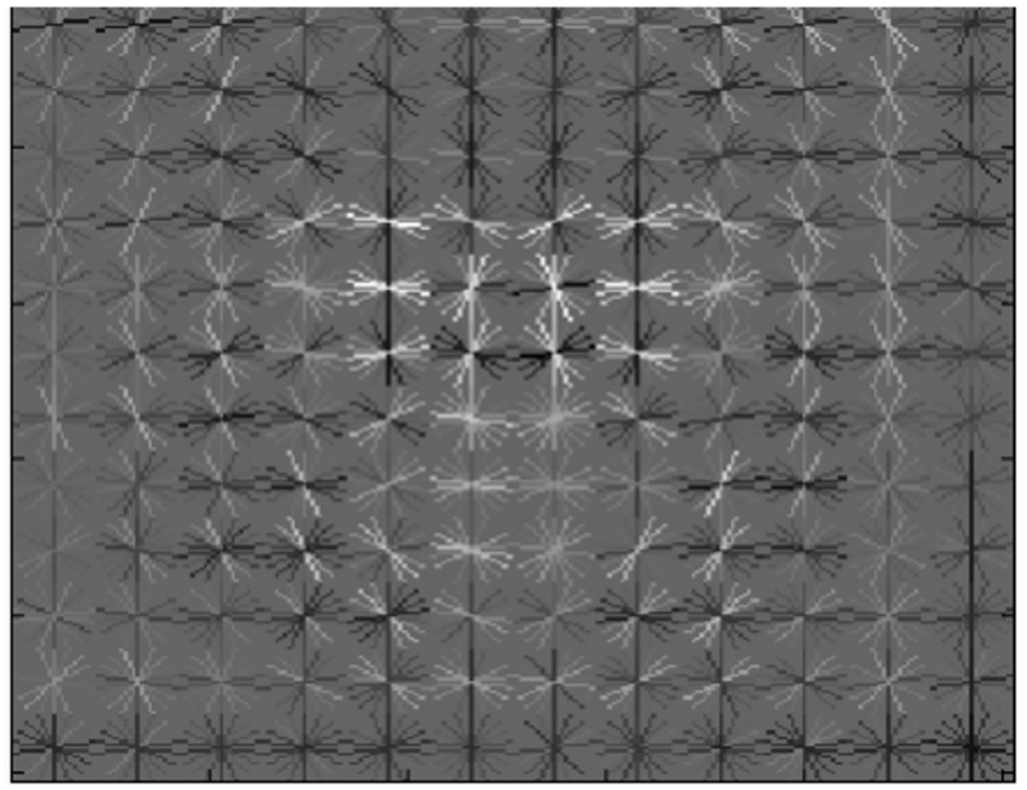




Model based Shapes

Deformable Part based Models (DPM)



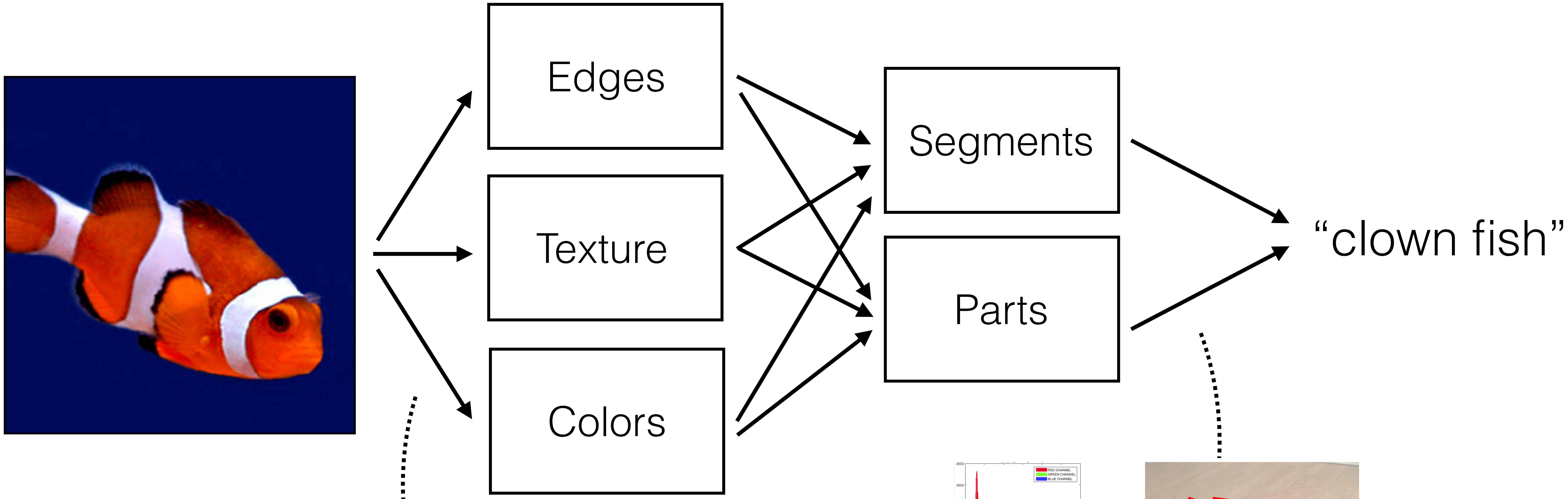


Histogram of Gradients (HOG)

Felzenszwalb et al. 2010.
Dalal and Triggs, 2005.
Beis and Lowe, 1997.

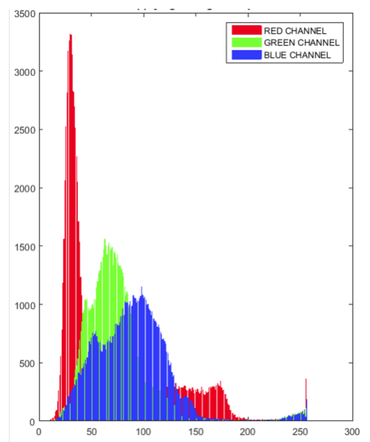
Example from CS331B: Representation Learning in Computer Vision

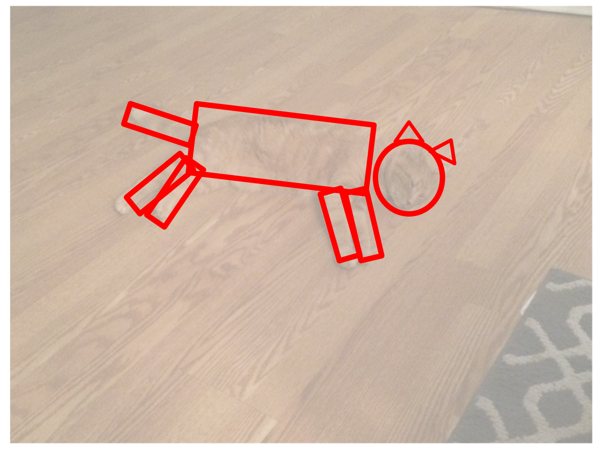
Traditional CV Pipeline



Feature extractors

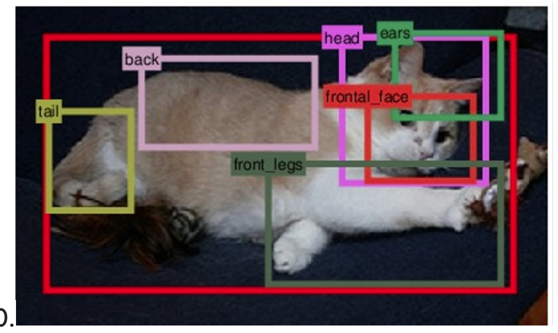
Color Histograms

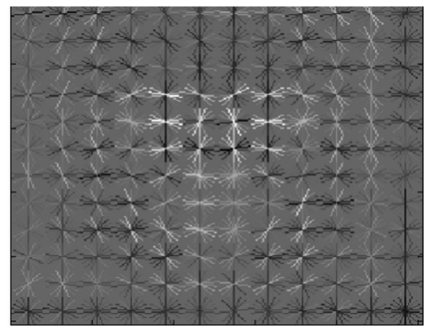




Model based Shapes

Deformable Part based Models (DPM)



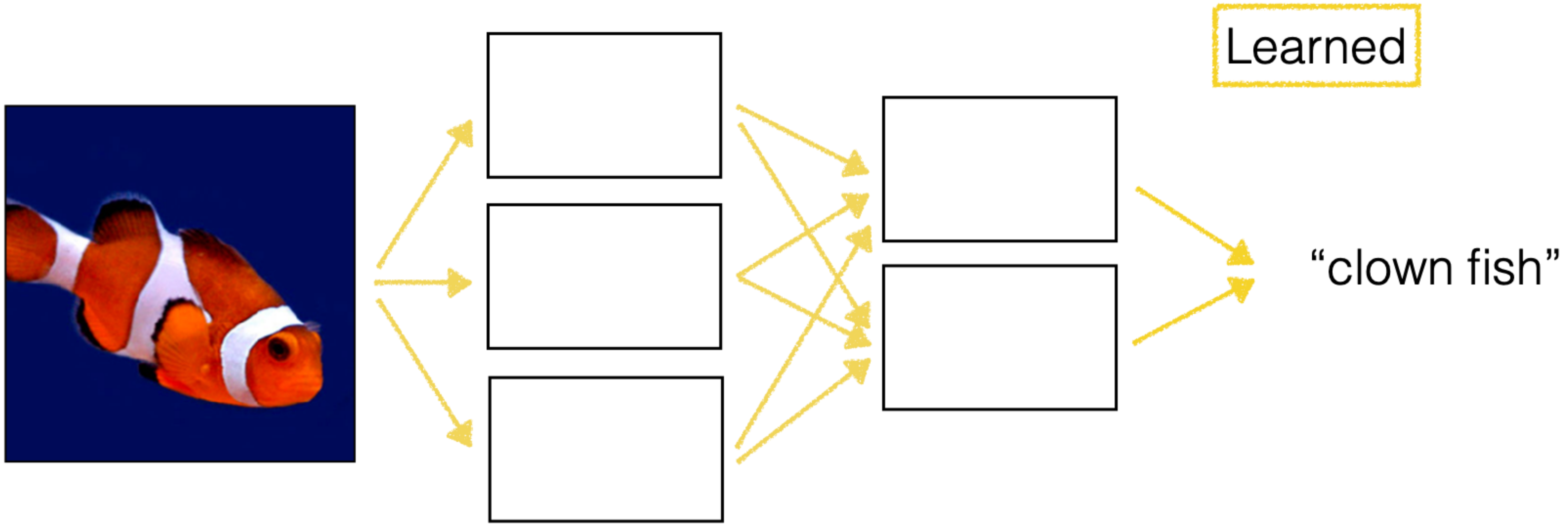


Histogram of Gradients (HOG)

Felzenszwalb et al. 2010.
Dalal and Triggs, 2005.
Beis and Lowe, 1997.

From MIT 6.8300/6.8301: Advances in Computer Vision

Learned CV Pipeline



Today's Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by
unsupervised learning and **self-supervised** learning?

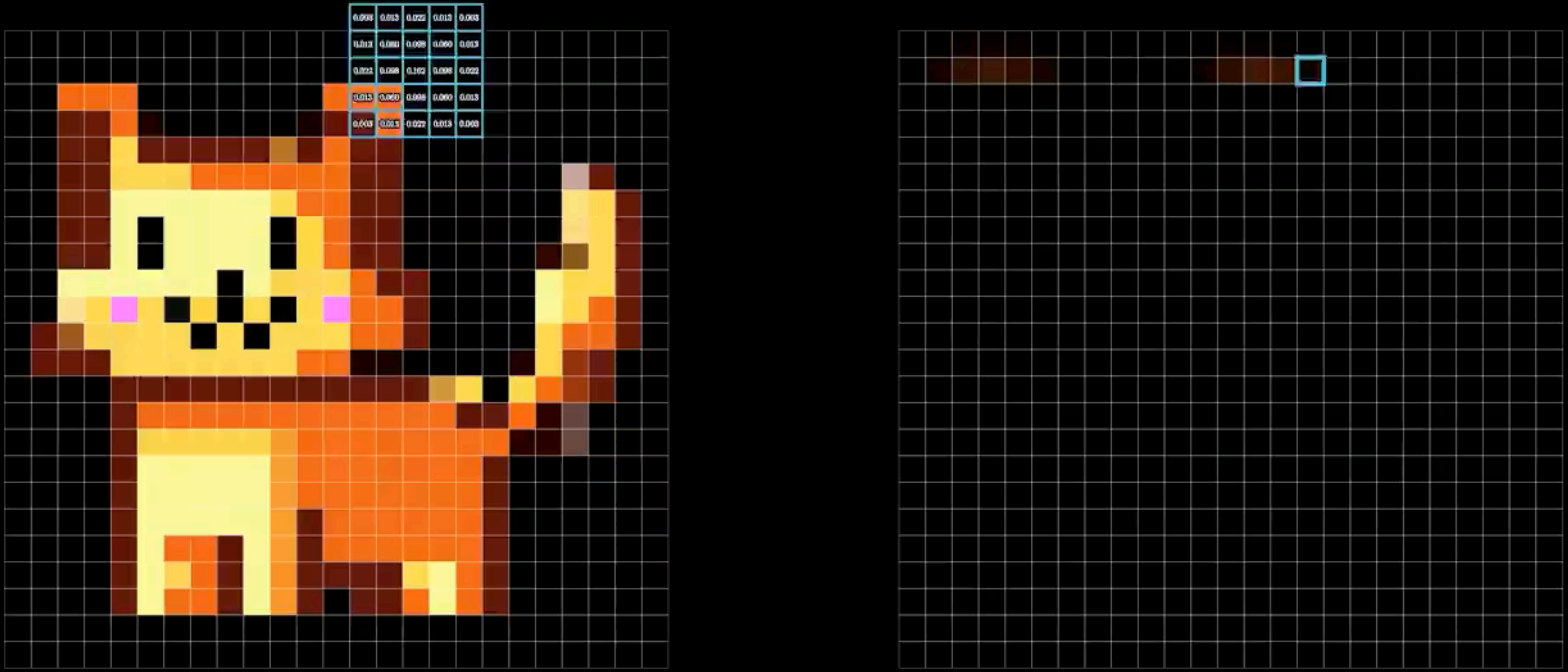
Today's Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

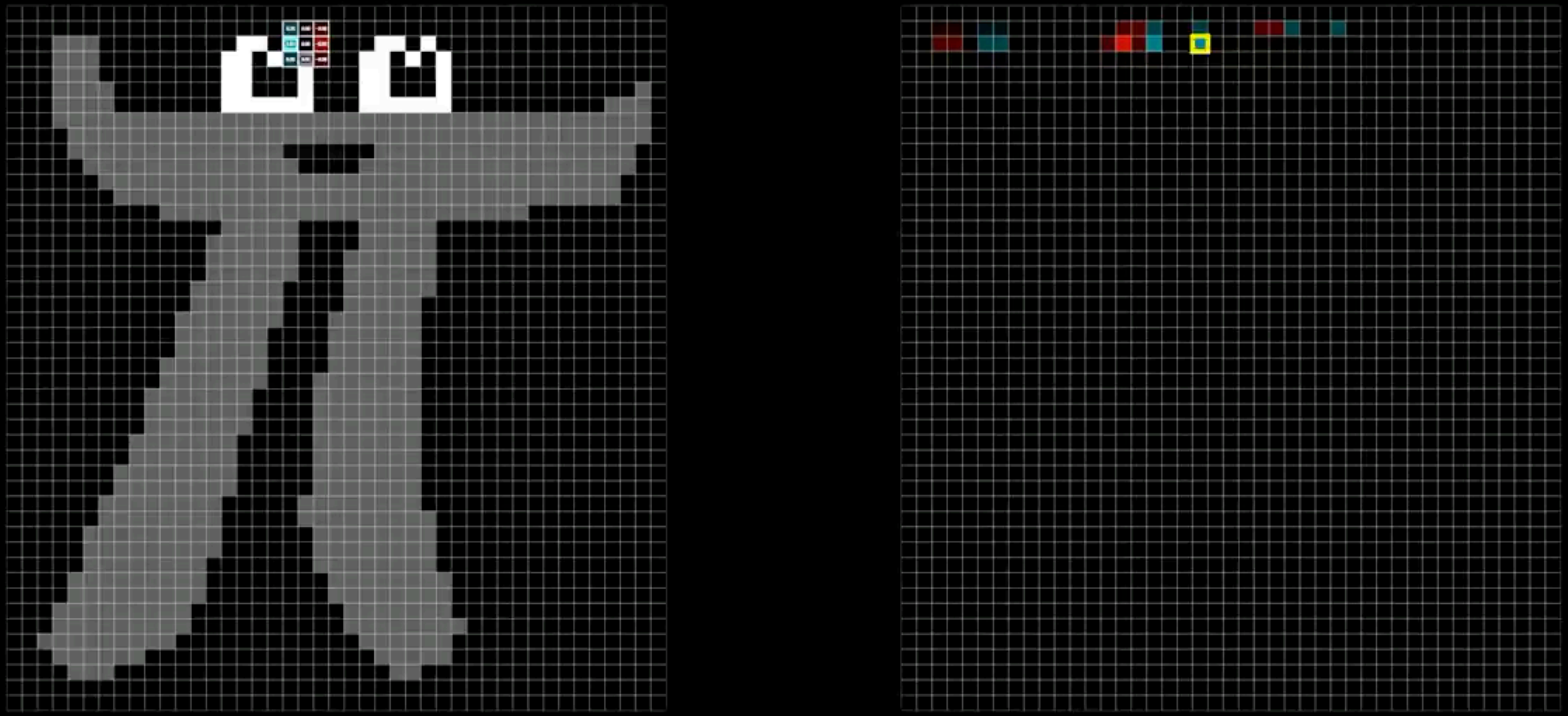
How do we train a representation by
unsupervised learning and **self-supervised** learning?

Convolutional Neural Network



Video from ThreeBlue1Brown: https://www.youtube.com/watch?v=KuXjwB4LzSA&t=738s

0.25	0.00	−0.25
0.50	0.00	−0.50
0.25	0.00	−0.25

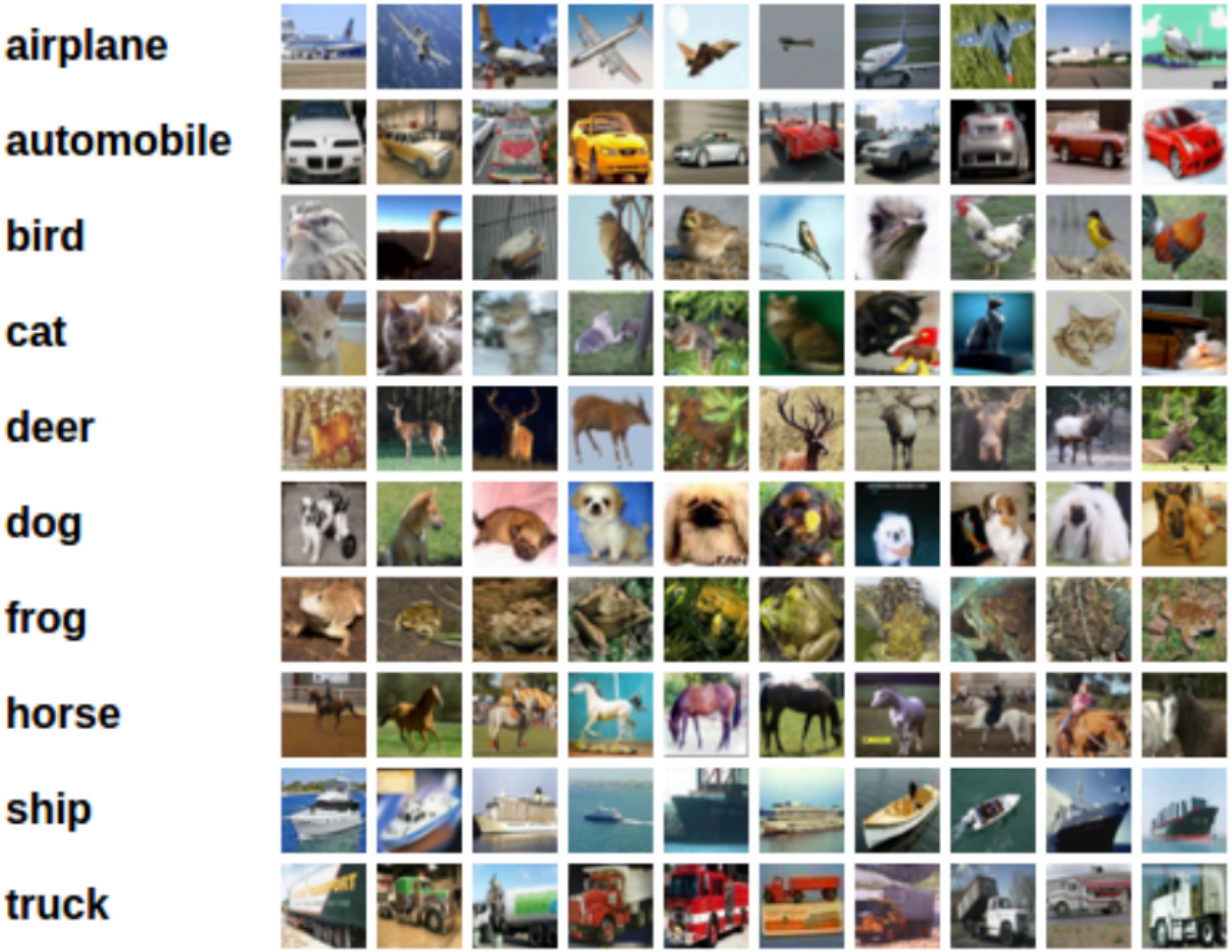


Video from ThreeBlue1Brown: https://www.youtube.com/watch?v=KuXjwB4LzSA&t=738s

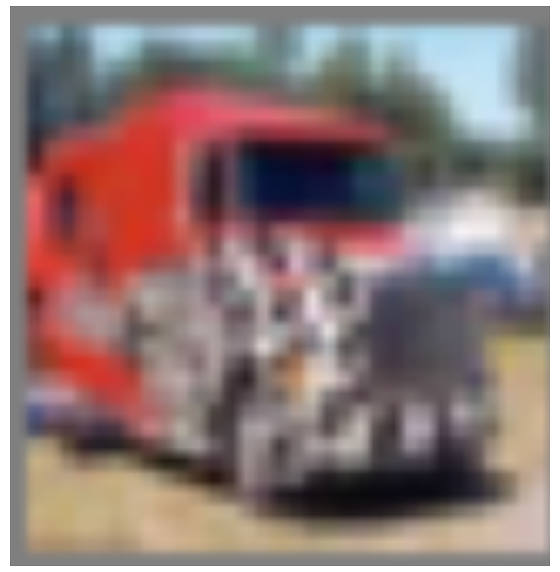
Training a CNN Classifier

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

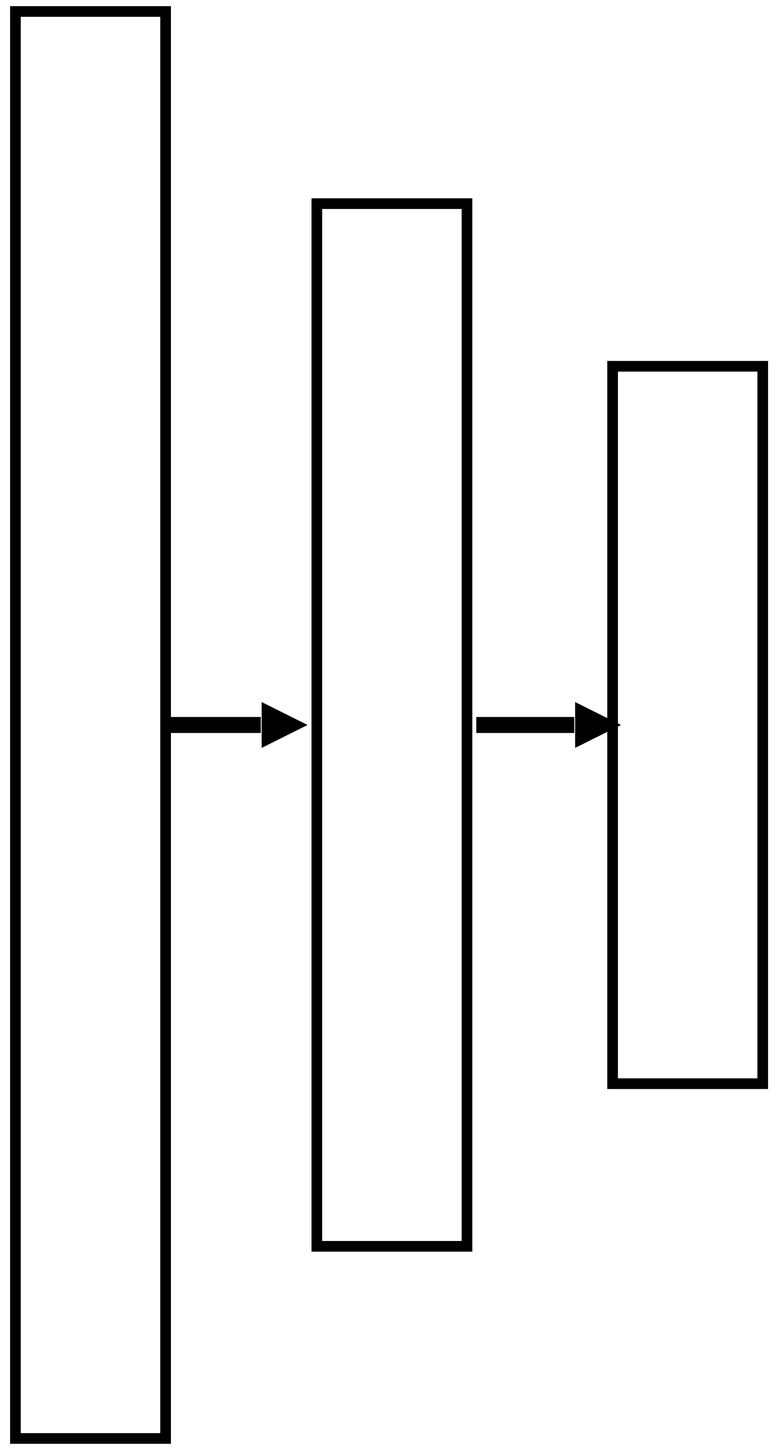
CIFAR-10 Dataset



CIFAR-10 Dataset



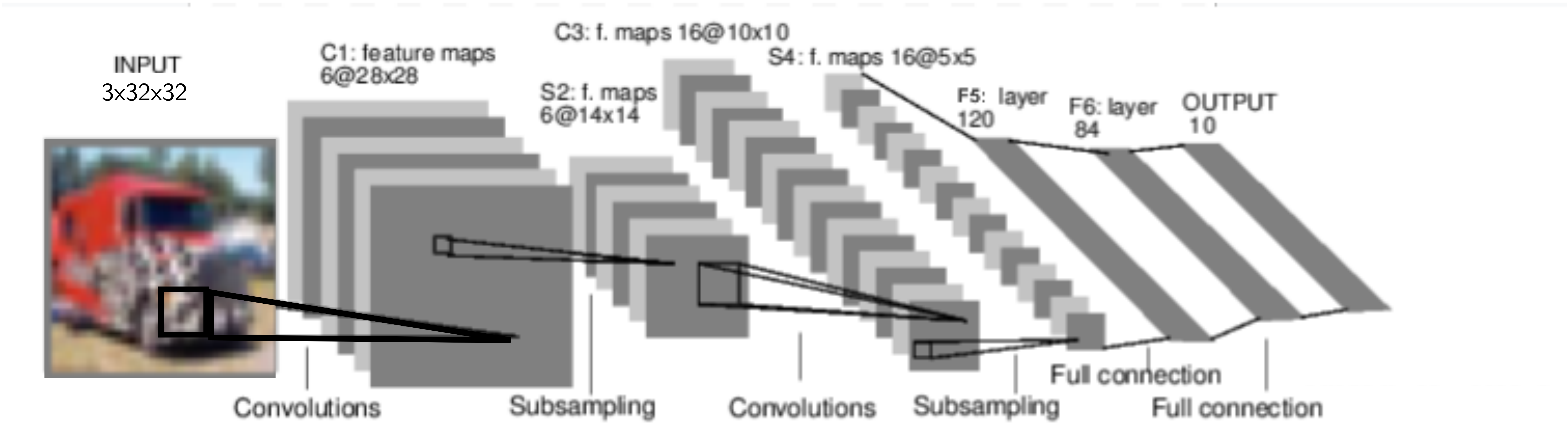
Input: Image



0.0
0.1
0.0
0
0.07
0.01
0.0
0.01
0.8 Truck!
0.01

Output: Class Probabilities

Model Architecture

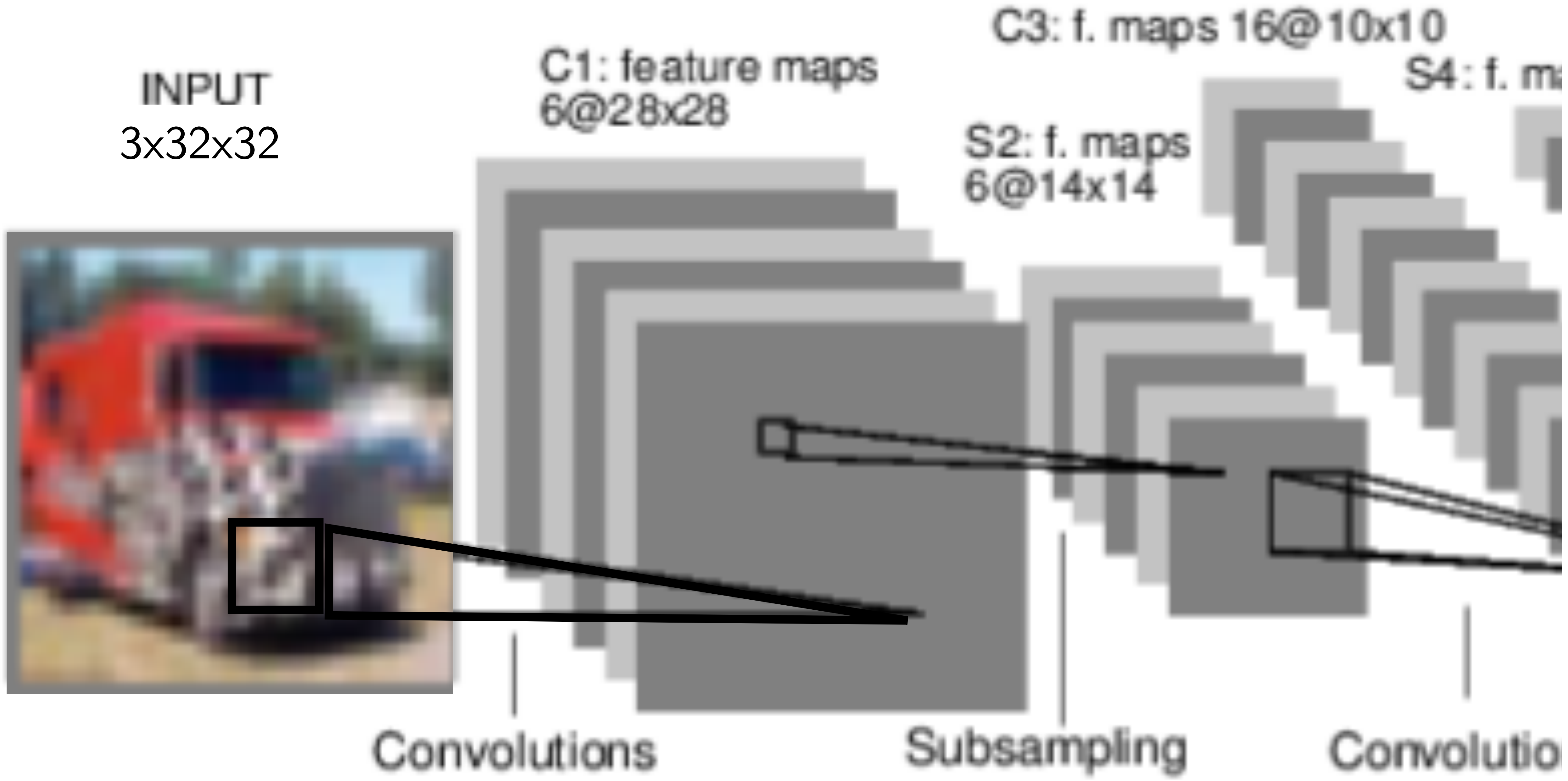


Input:
Image

Output:
Logits for class
(1-10)

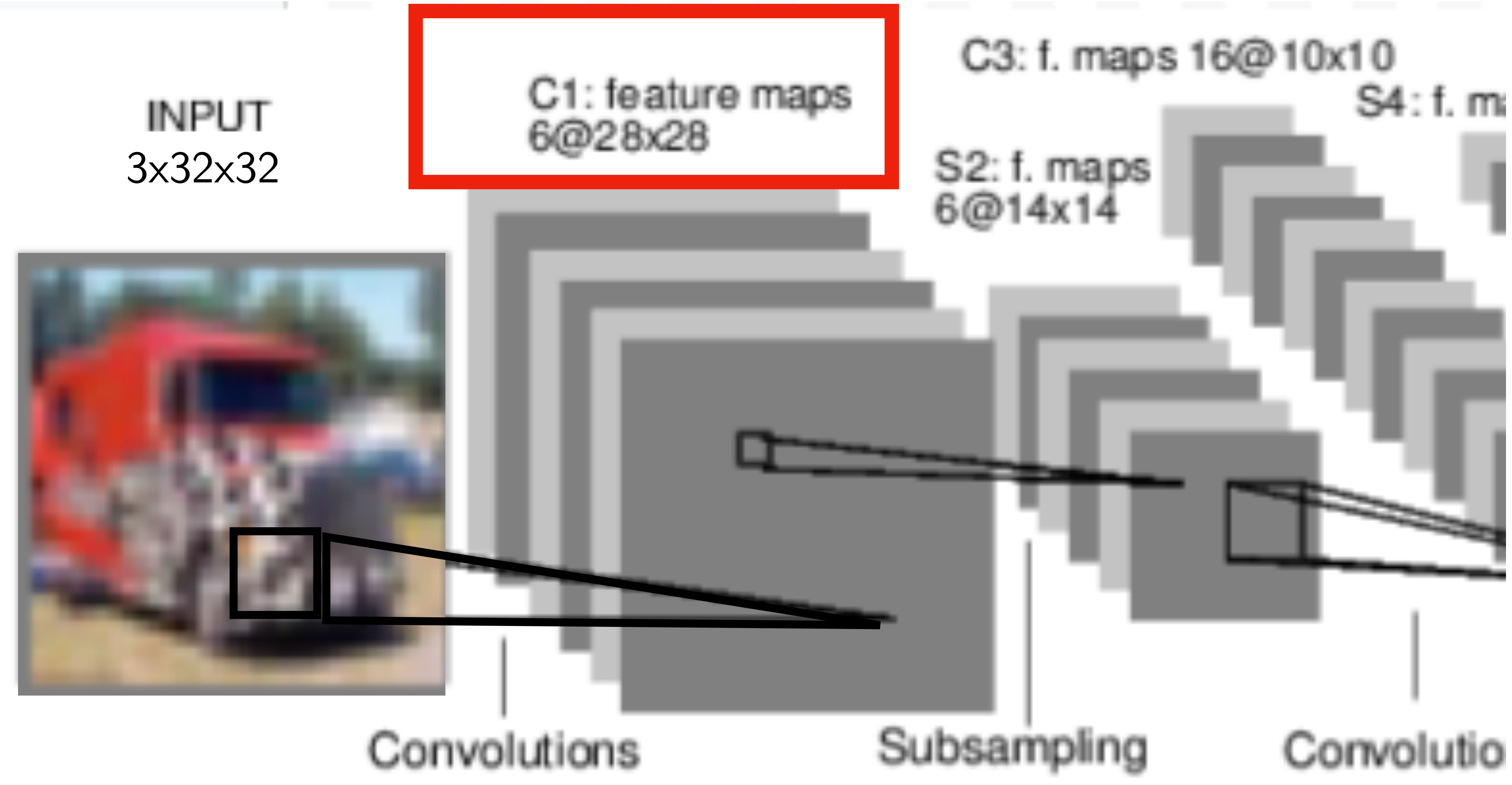
classes = ('plane', 'car', 'bird', 'cat',
'deer', 'dog', 'frog', 'horse', 'ship', 'truck')

Model Architecture




```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
```

Model Architecture



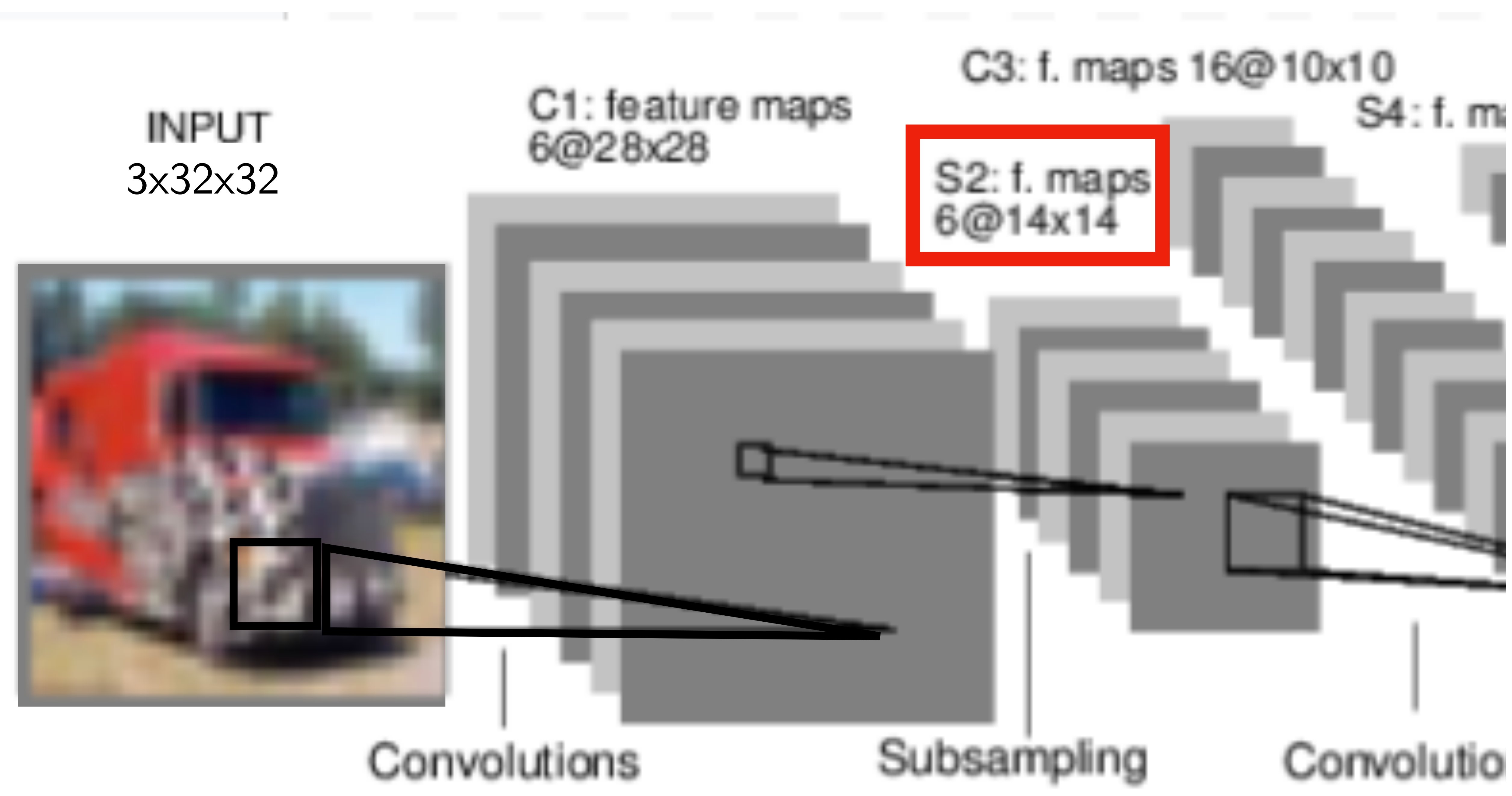

```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
```

Kernel Size

Input Channel

Output Channel

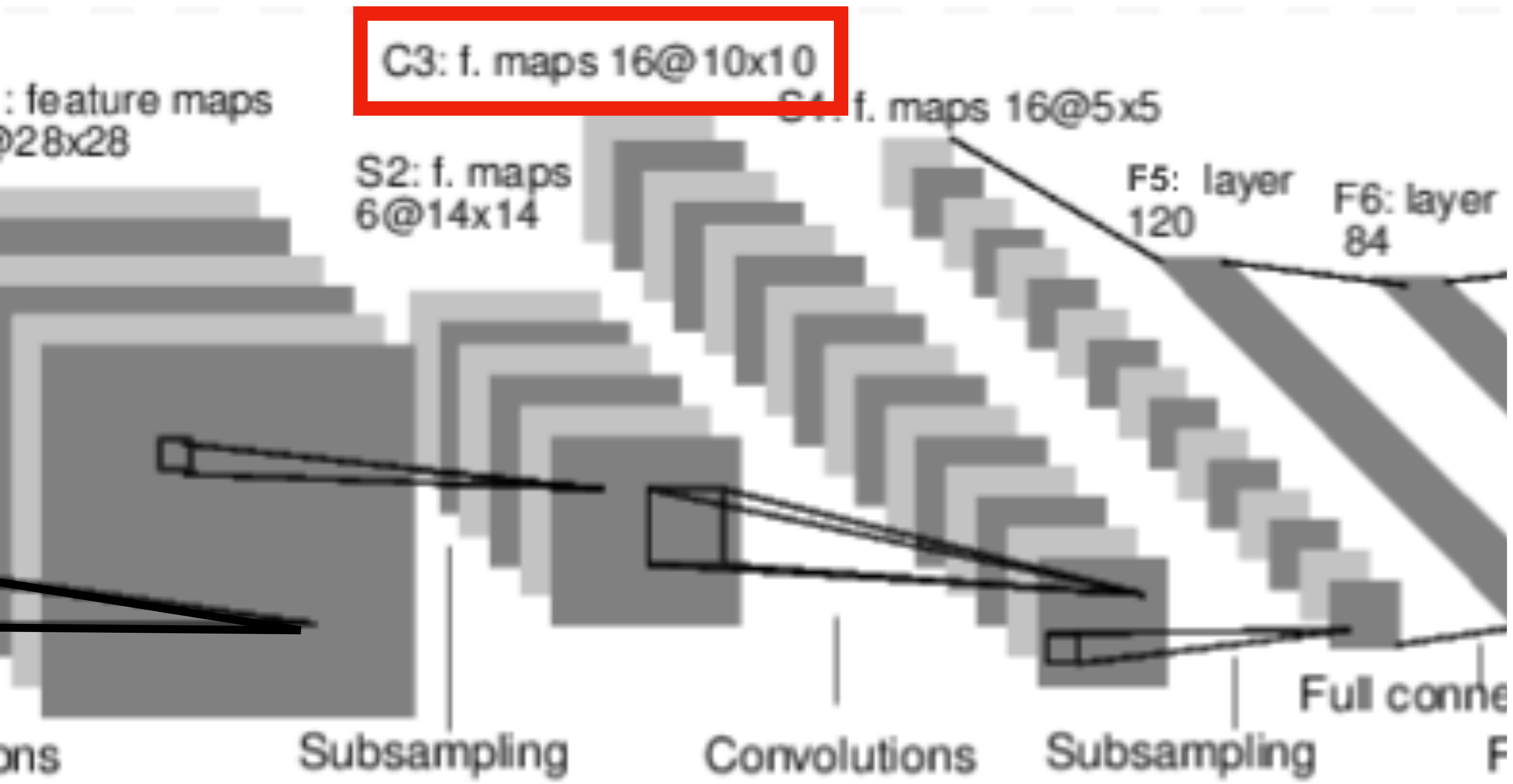
Model Architecture




```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
```

Kernel Size

Model Architecture



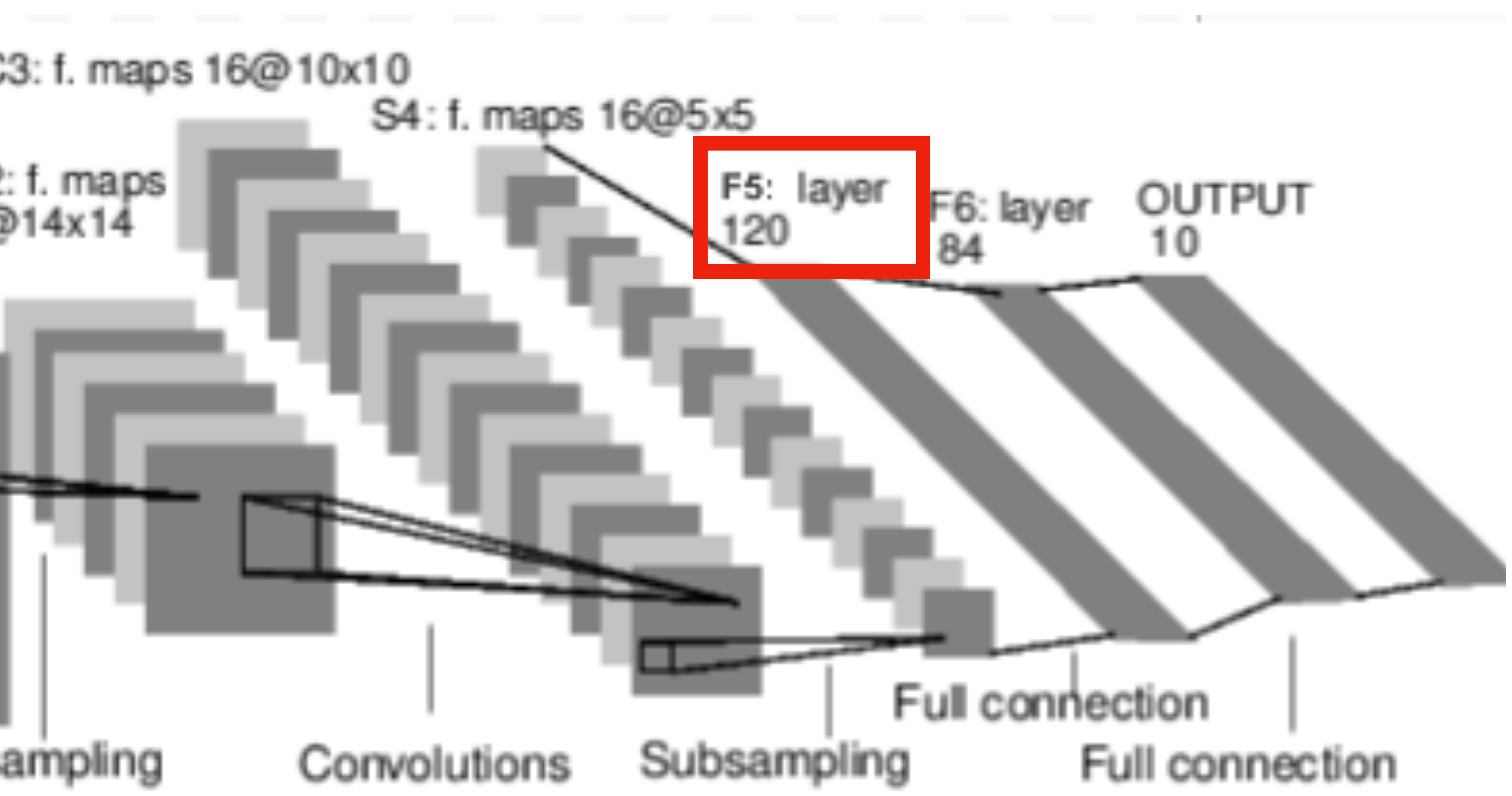

```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
```

Input Channel

Output Channel

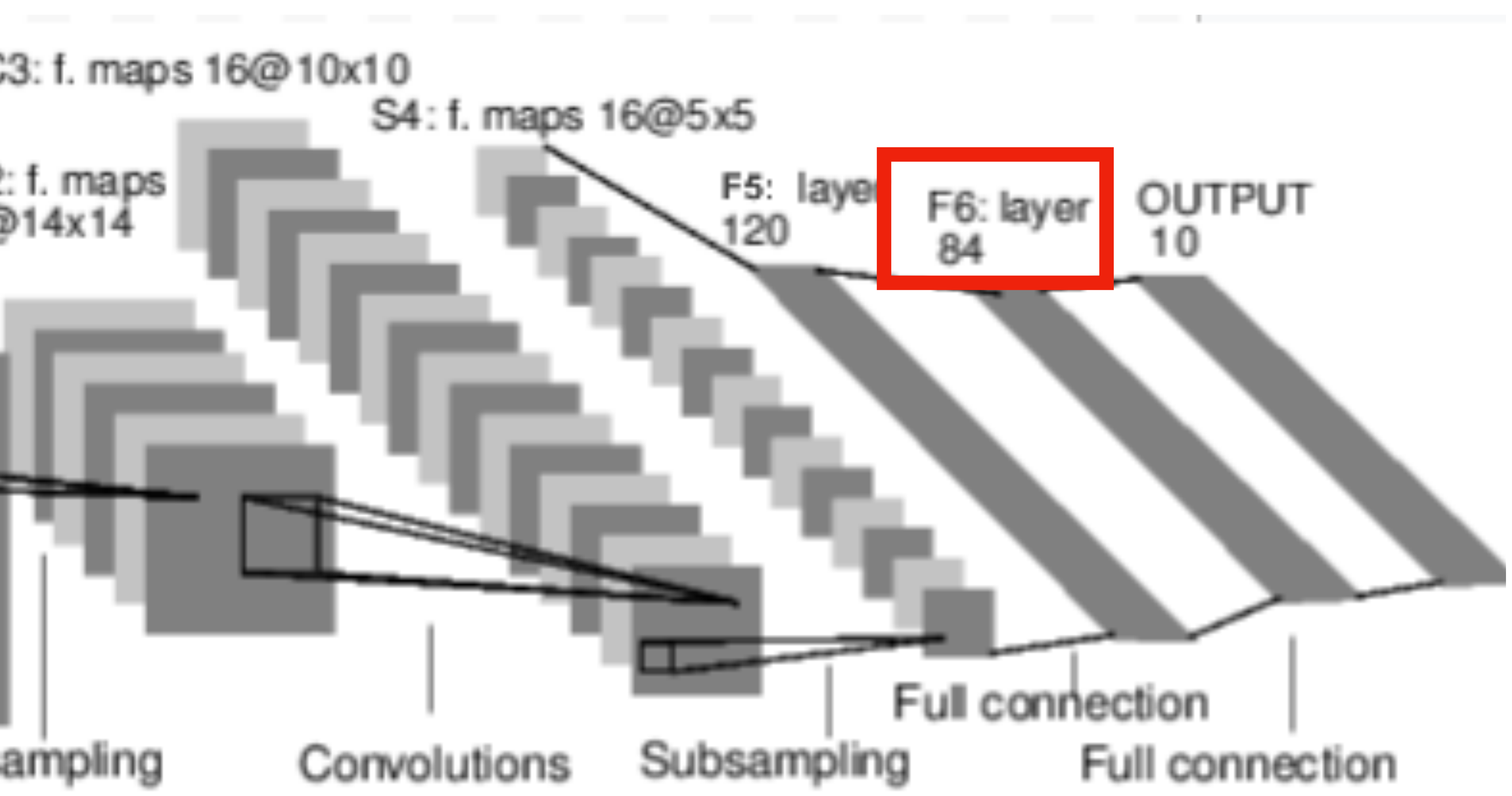
Kernel Size

Model Architecture



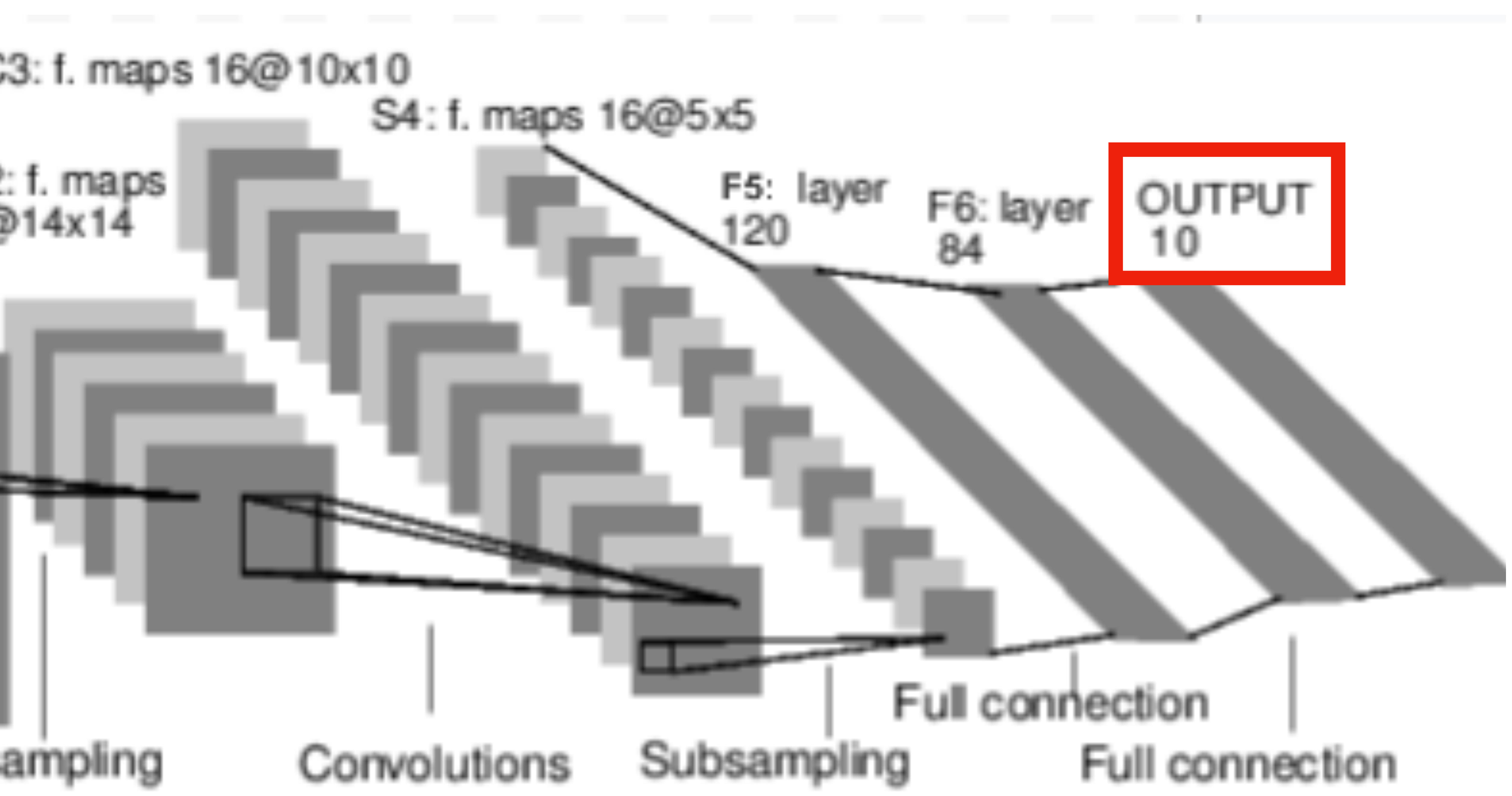

```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5,
                             120)
```

Model Architecture




```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5,
                             120)
        self.fc2 = nn.Linear(120, 84)
```

Model Architecture




```
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5,
                             120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)
```

Go through rest of the tutorial, train a classifier, and try to improve accuracy (Hint: Increase the width of the network)

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

Visualizing and Understanding CNNs

[Zeiler and Fergus, 2014]

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

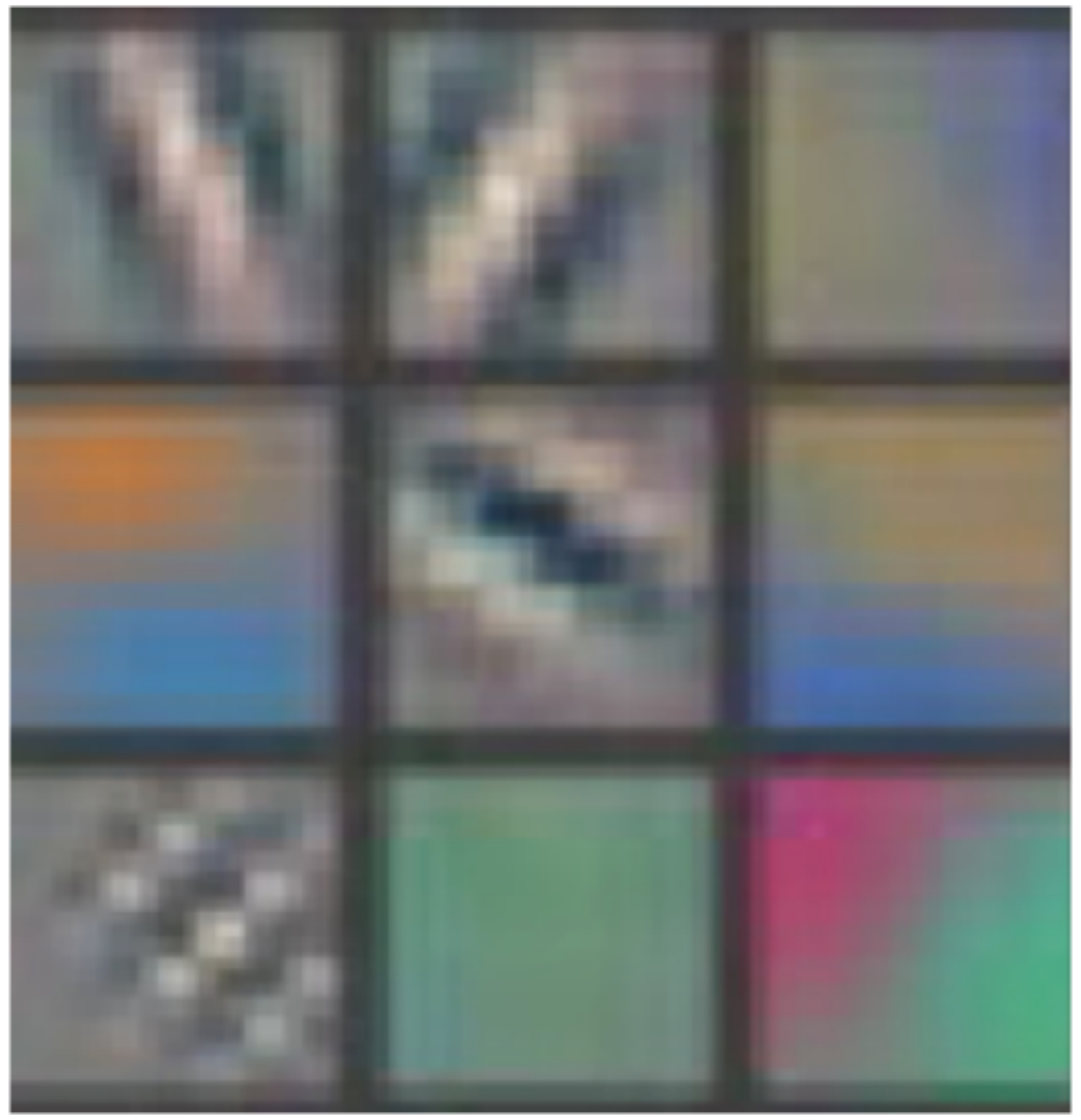
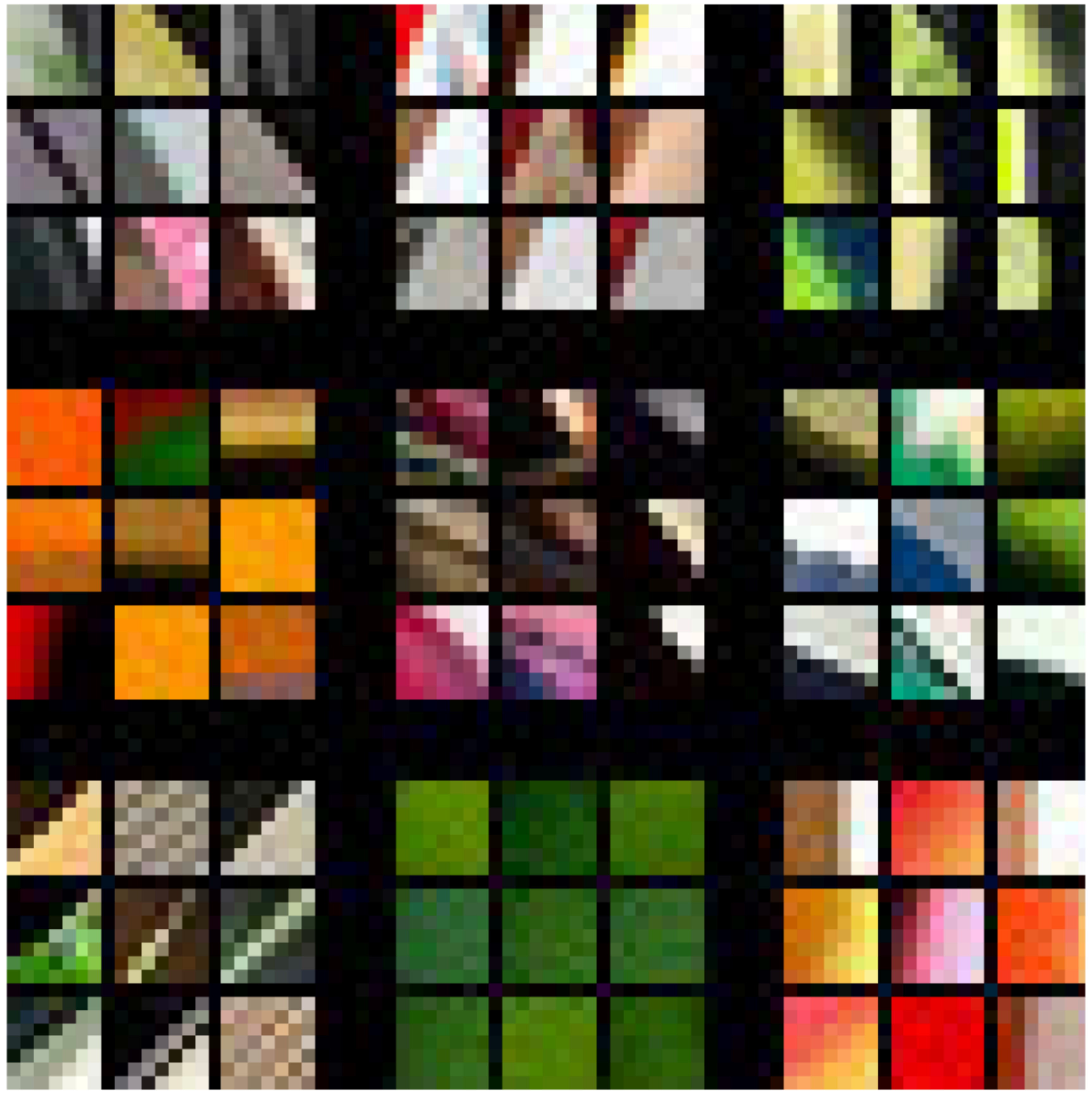
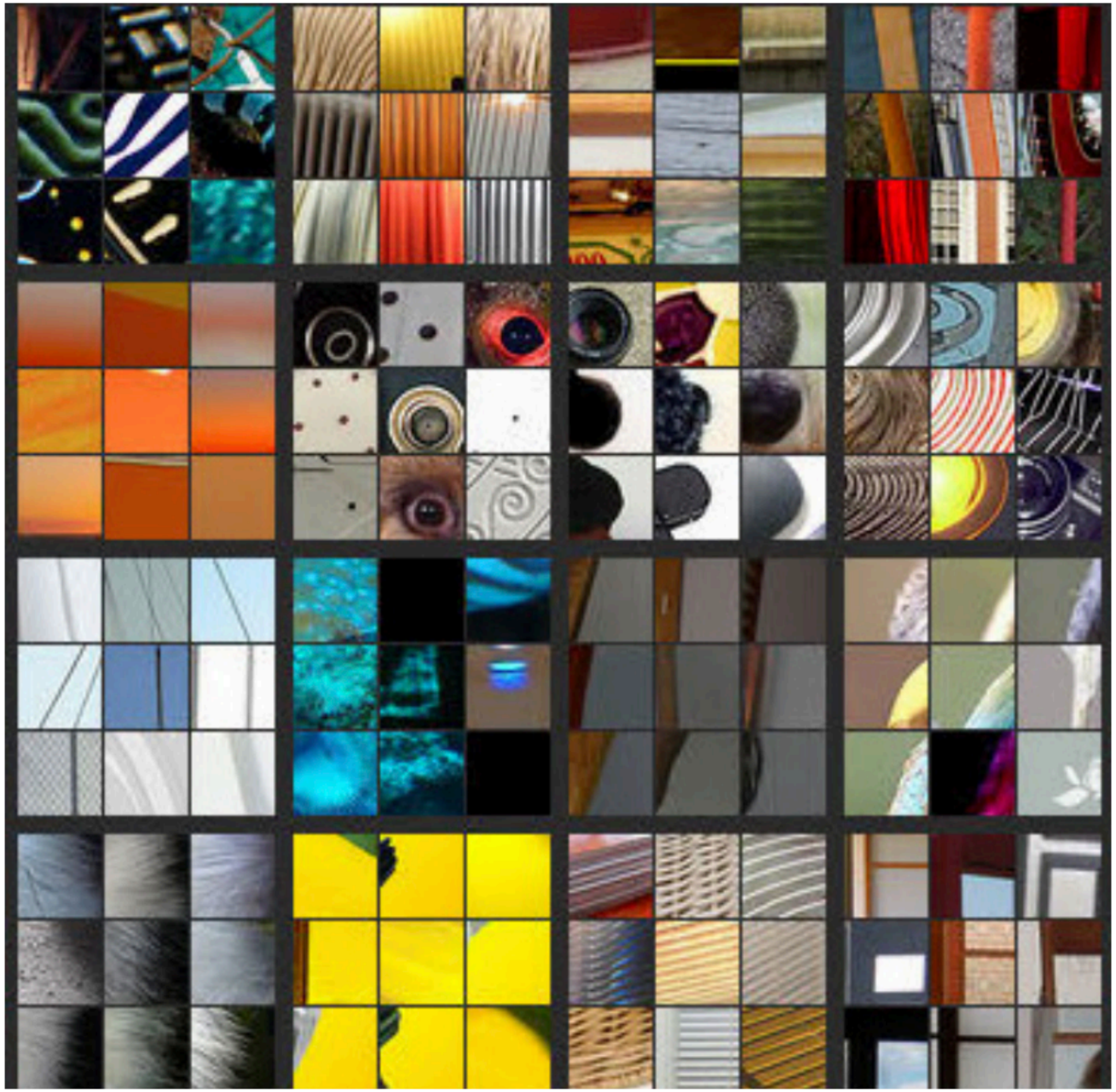


Image patches that activate each of the **layer 1** filters most strongly



[Zeiler and Fergus, 2014]

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



[Zeiler and Fergus, 2014]

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

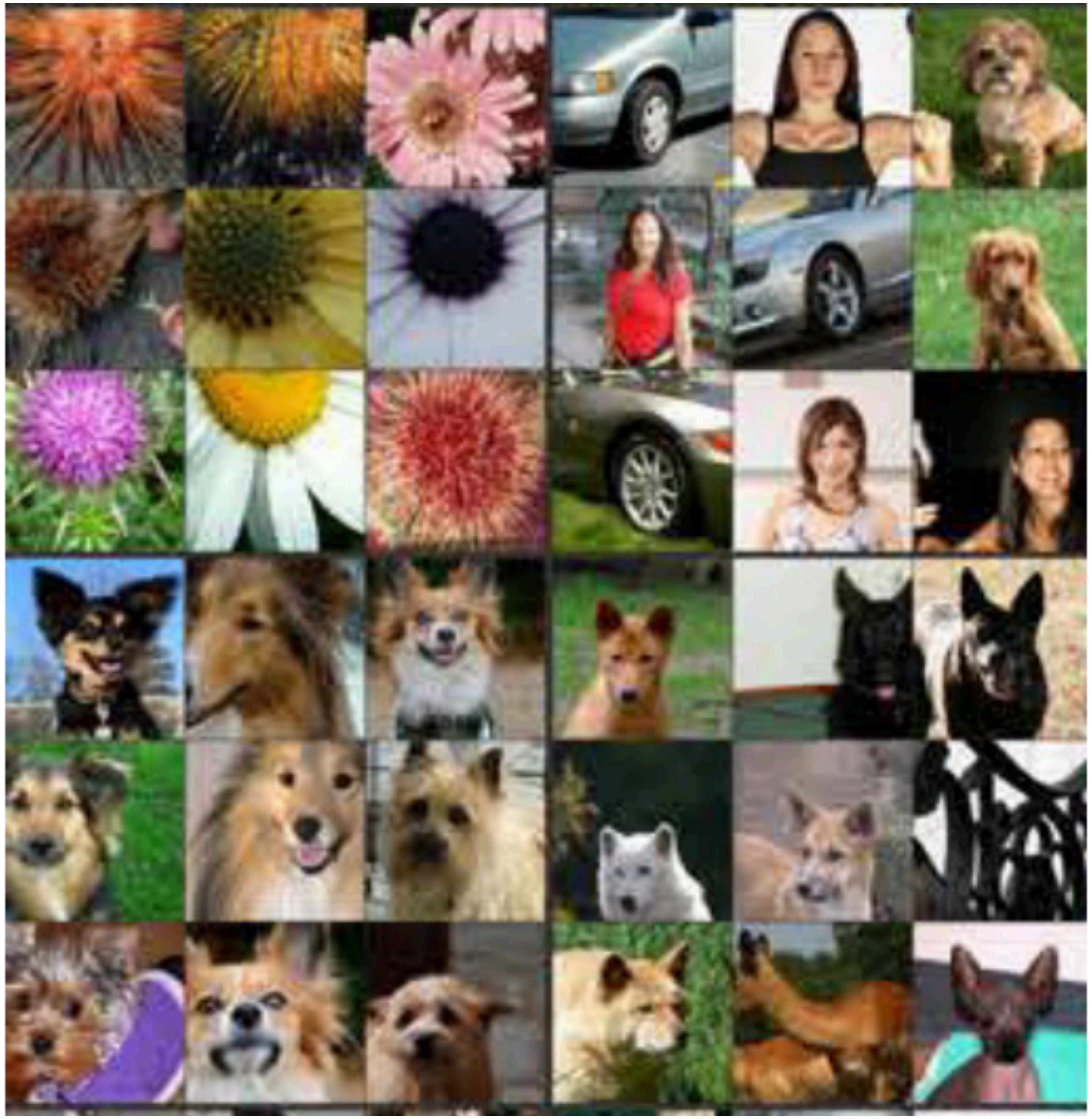
From MIT 6.8300/6.8301: Advances in Computer Vision

[Zeiler and Fergus, 2014]

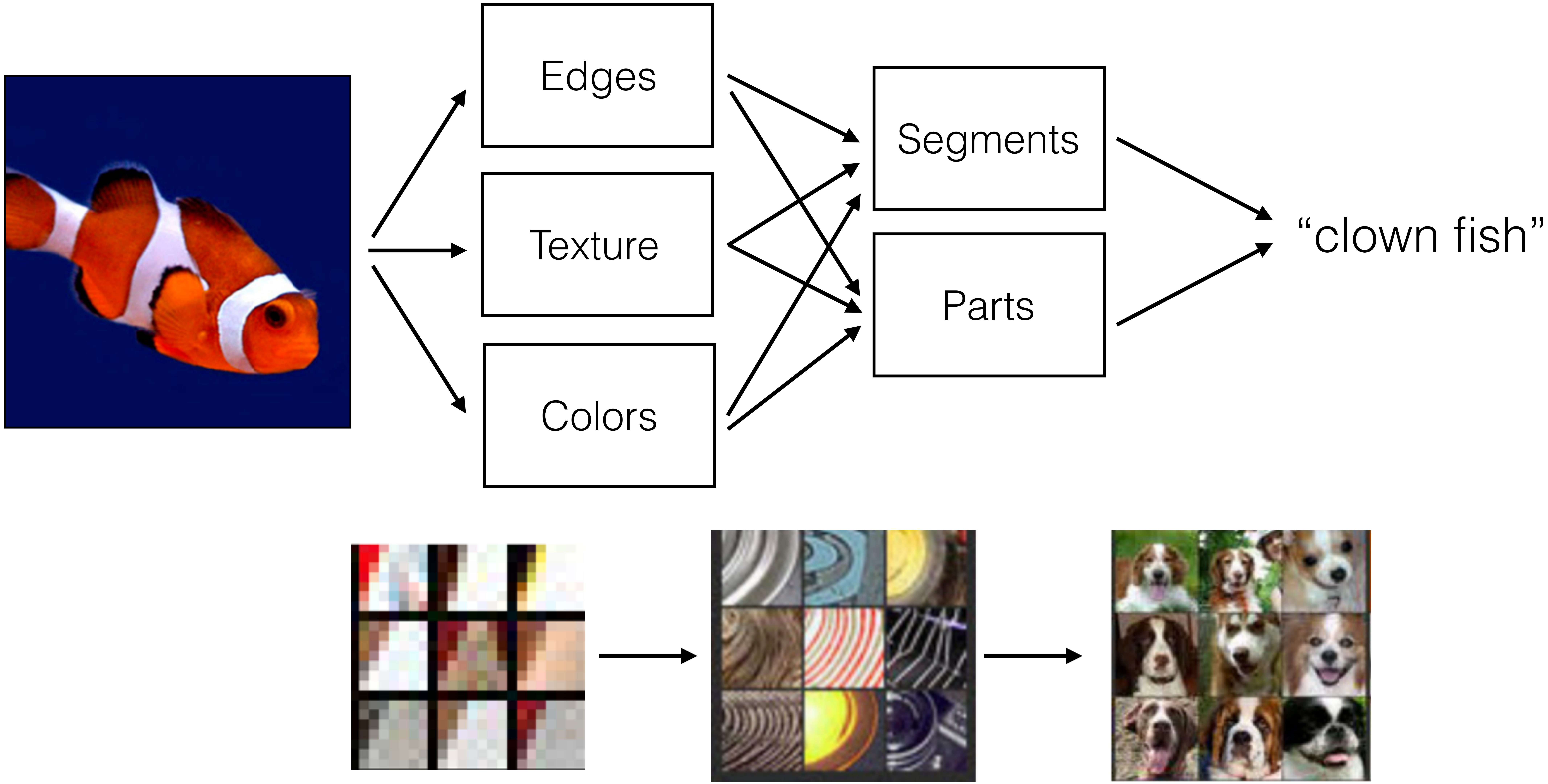
Image patches that activate several of the **layer 4** neurons most strongly

[Zeiler and Fergus, 2014]

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



CNNs *learned* the classical visual recognition pipeline!



From MIT 6.8300/6.8301: Advances in Computer Vision

Today's Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by **unsupervised** learning and **self-supervised** learning?

Today's Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by
unsupervised learning and **self-supervised** learning?

Supervised object recognition

Learner

"Fish"

image X

label Y

Supervised object recognition

Learner

"Fish"

image X

label Y

From MIT 6.8300/6.8301: Advances in Computer Vision

Supervised object recognition

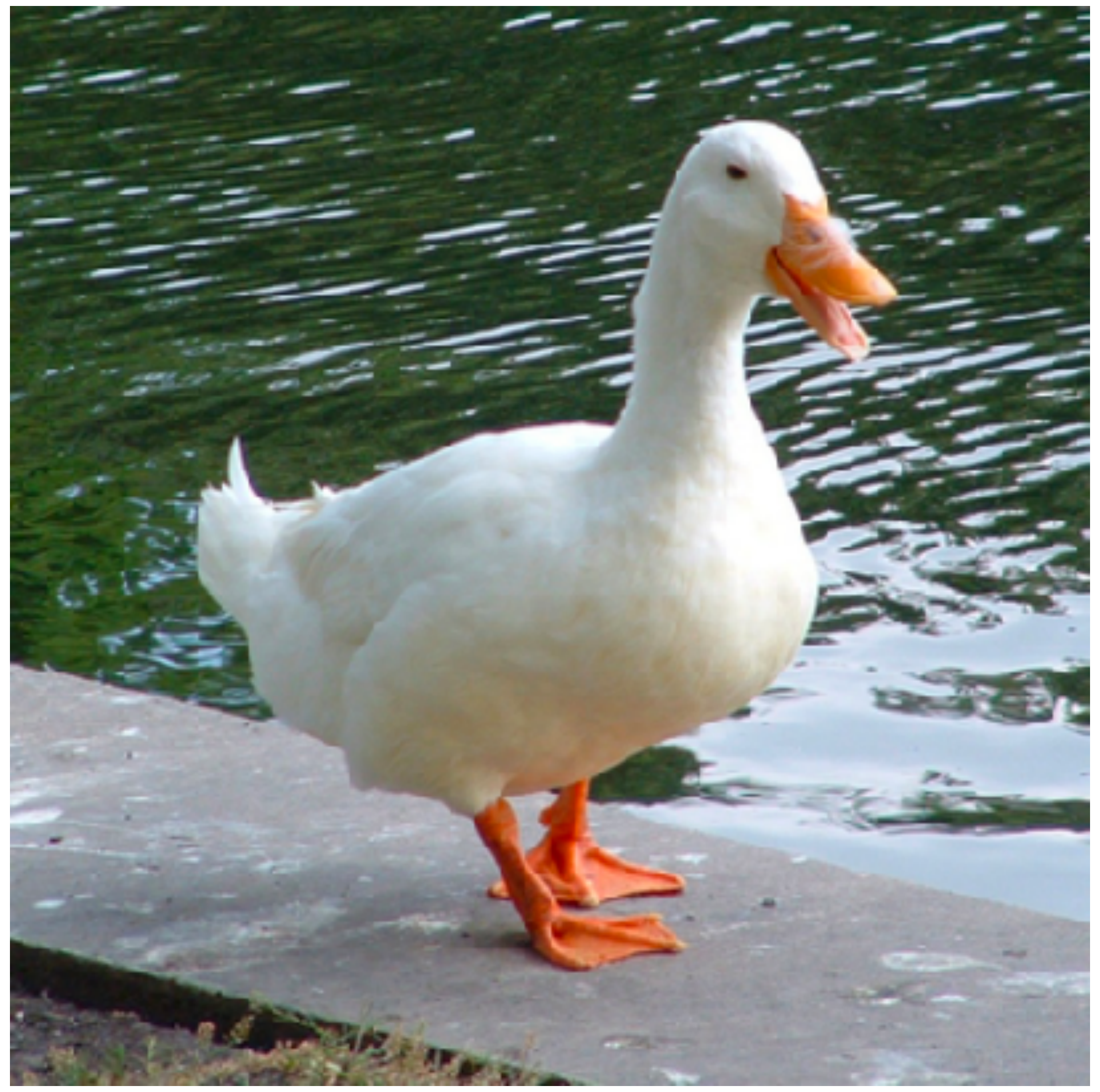
Learner

"Fish"

image X

label Y

Supervised object recognition



Learner

"Duck"

⋮

image X

label Y

The real world is not labeled!

Imagine a robot having to clean this kitchen.

How does it learn about all the objects in the scene?

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

Learning from labels

(aka **supervised learning**)

Training data

$$\{x^{(1)}, y^{(1)}\}$$

$$\{x^{(2)}, y^{(2)}\} \quad \rightarrow \quad \boxed{\text{Learner}} \quad \rightarrow \quad f : X \rightarrow Y$$

$$\{x^{(3)}, y^{(3)}\}$$

$$\dots$$

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

Learning without labels

(includes **unsupervised learning** and **reinforcement learning**)

Data

$\{x^{(1)}\}$

$\{x^{(2)}\}$ $\rightarrow$ Learner $\rightarrow$?

$\{x^{(3)}\}$

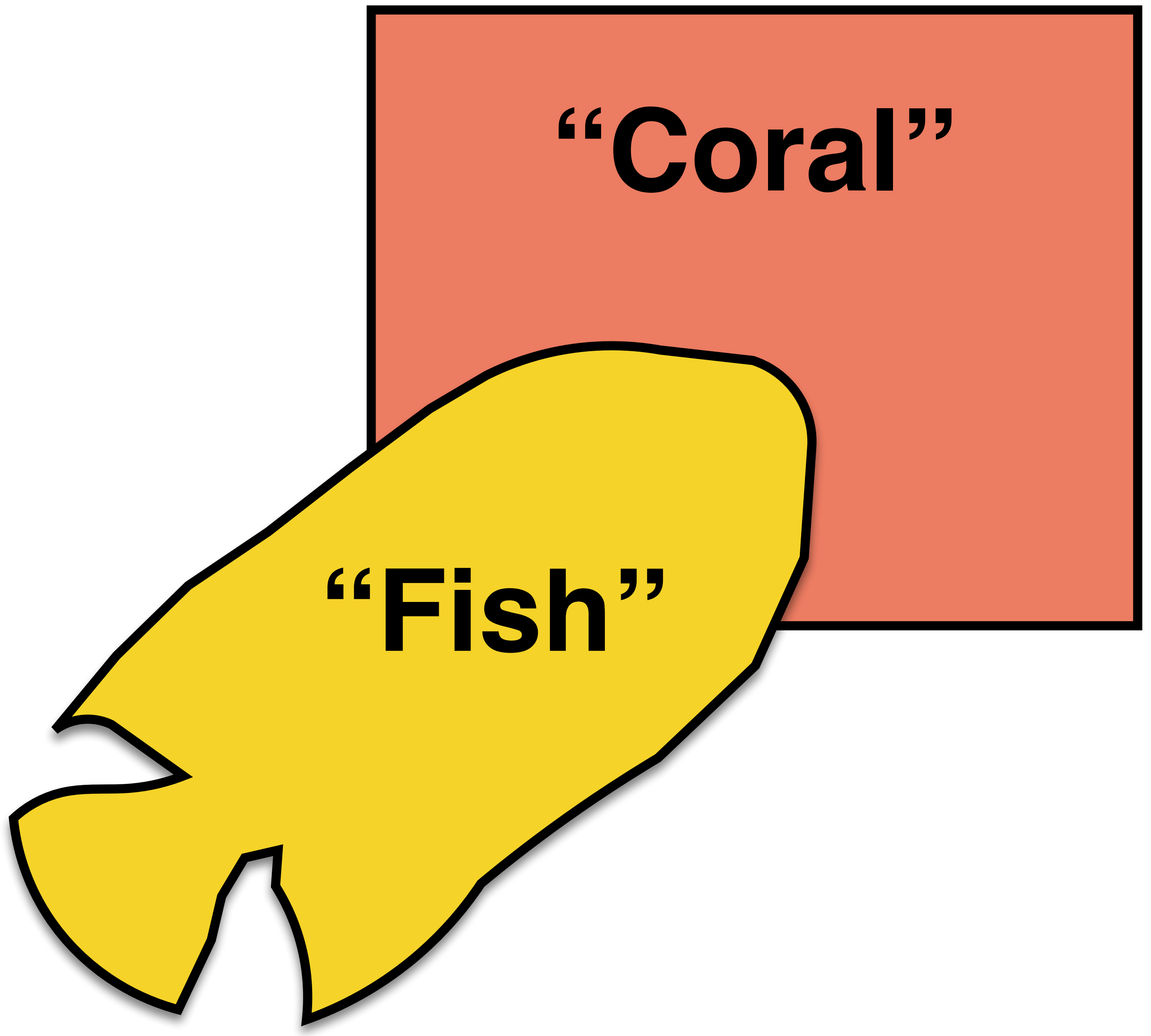
$\ldots$

Unsupervised Learning

Unsupervised Representation Learning

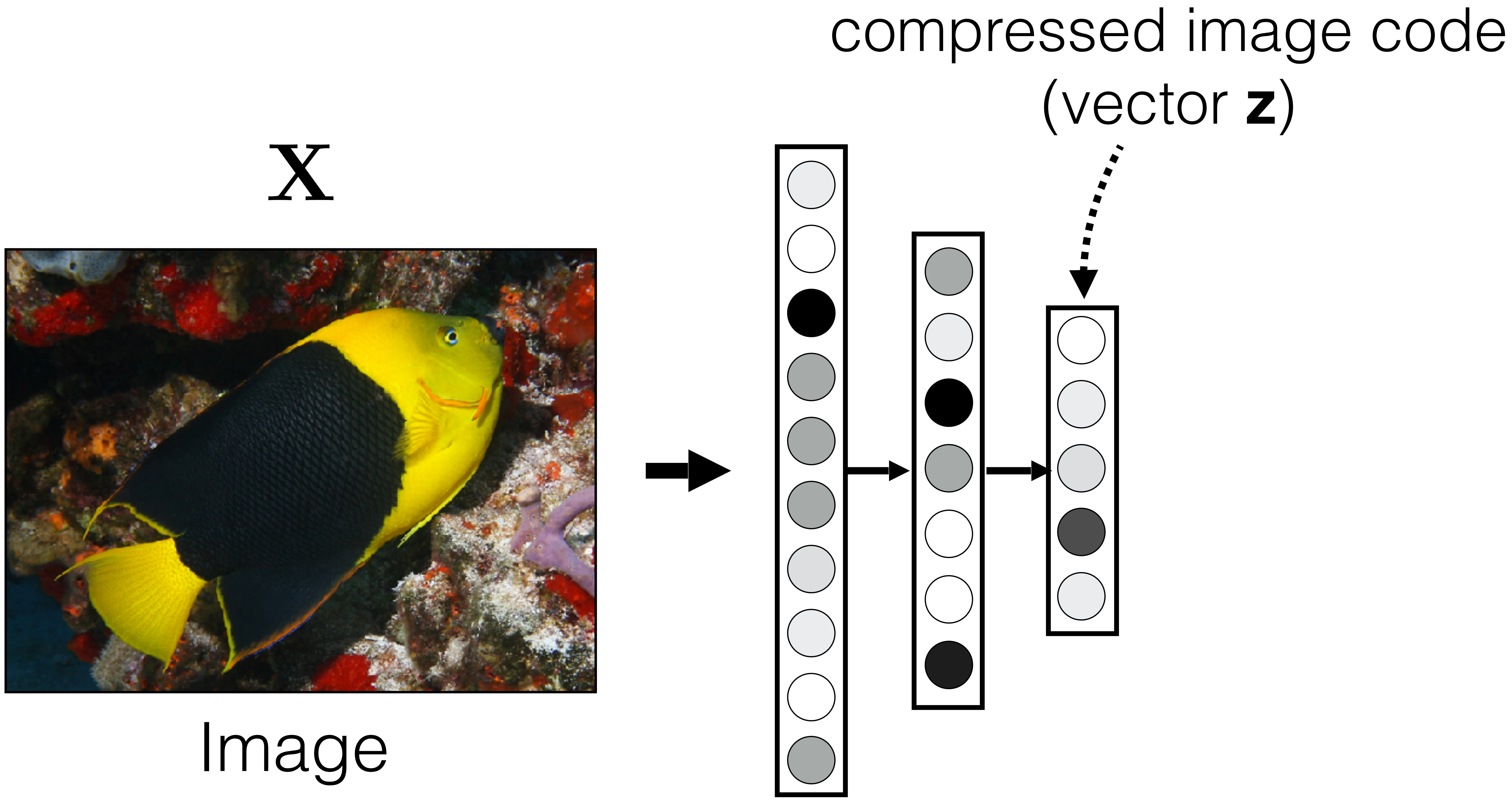
$\mathbf{x}$

Image



Compact mental representation

Unsupervised Representation Learning

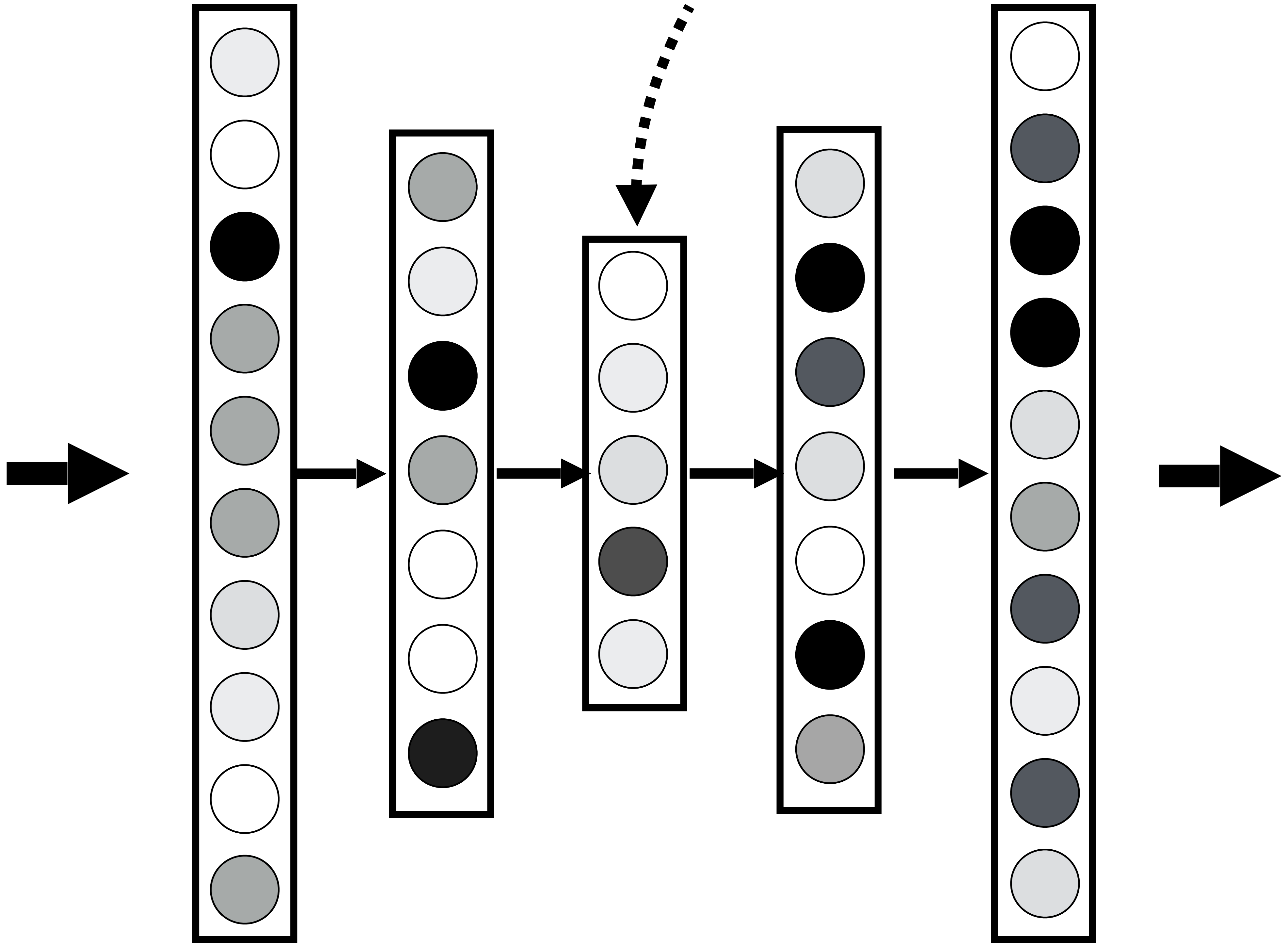


Unsupervised Representation Learning

compressed image code
(vector $\mathbf{z}$)

$\mathbf{x}$

Image



$\hat{\mathbf{x}}$

Reconstructed image

"Autoencoder"

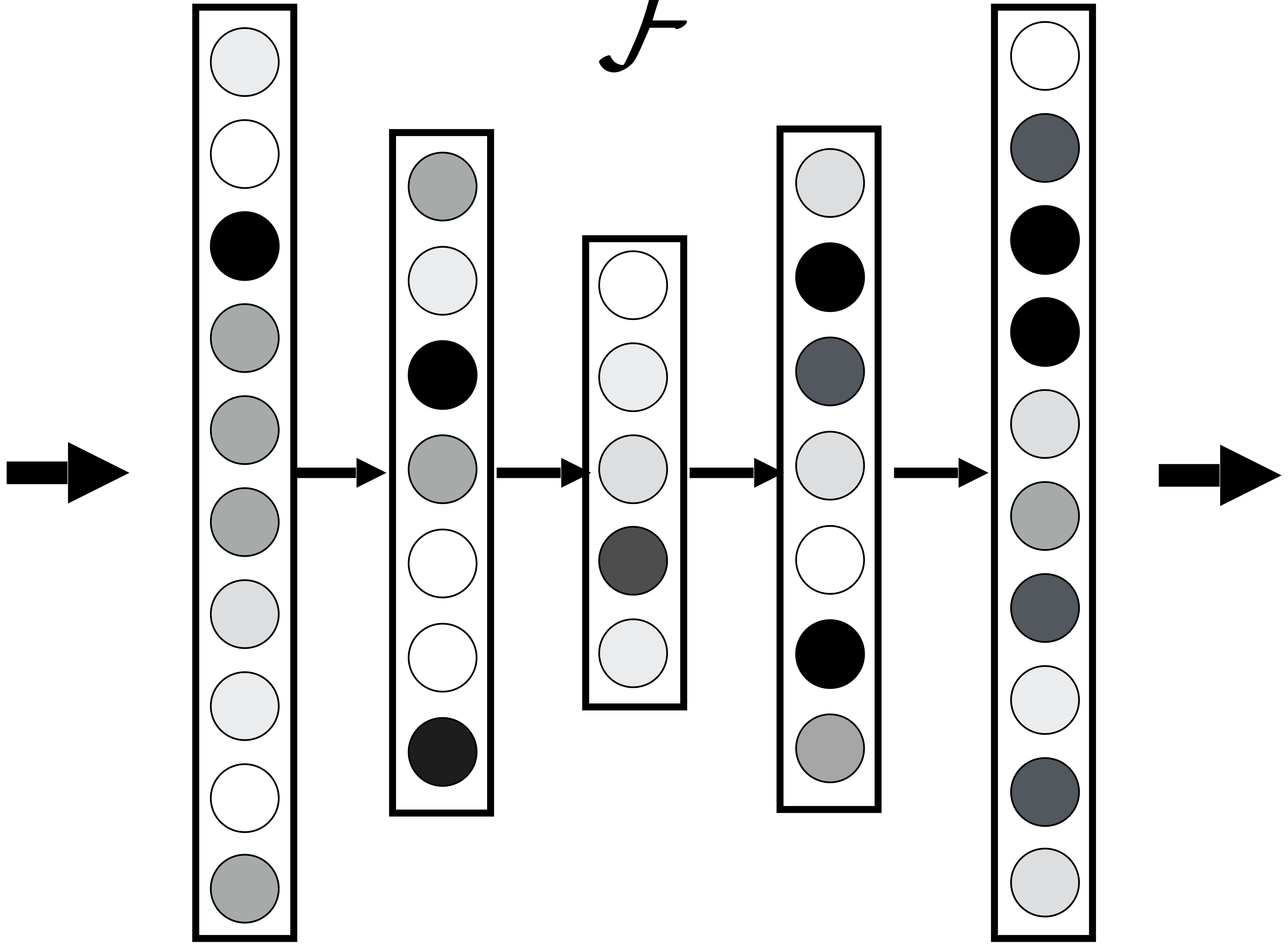
[e.g., Hinton & Salakhutdinov, Science 2006]

Autoencoder

$\mathbf{X}$

Image

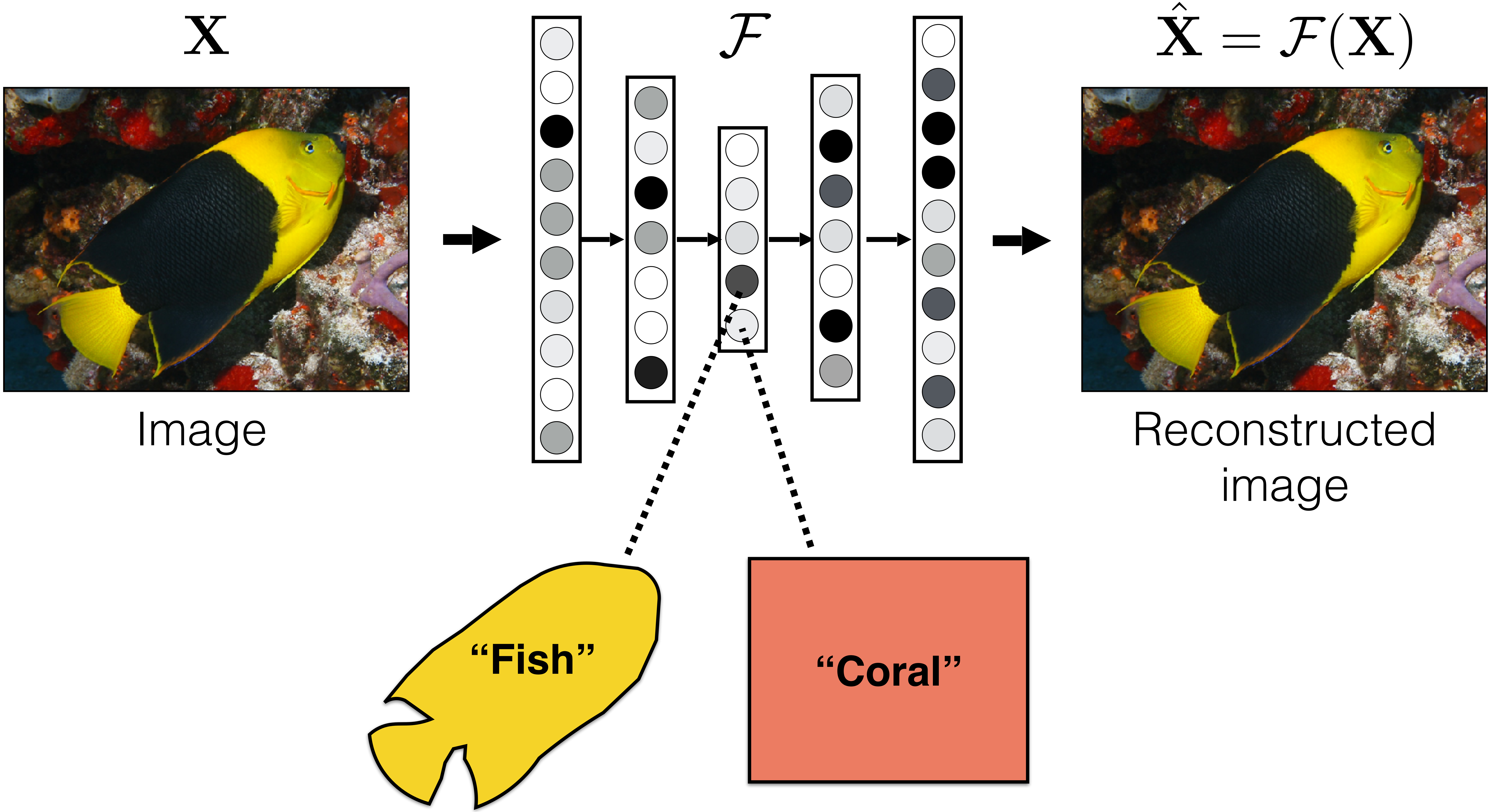
$\mathcal{F}$



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

Reconstructed image

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



$\mathbf{X}$
Image
$\mathcal{F}$
$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{X})$
Reconstructed image
"Fish"
"Coral"

Autoencoder

Data

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

Learner

Objective

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

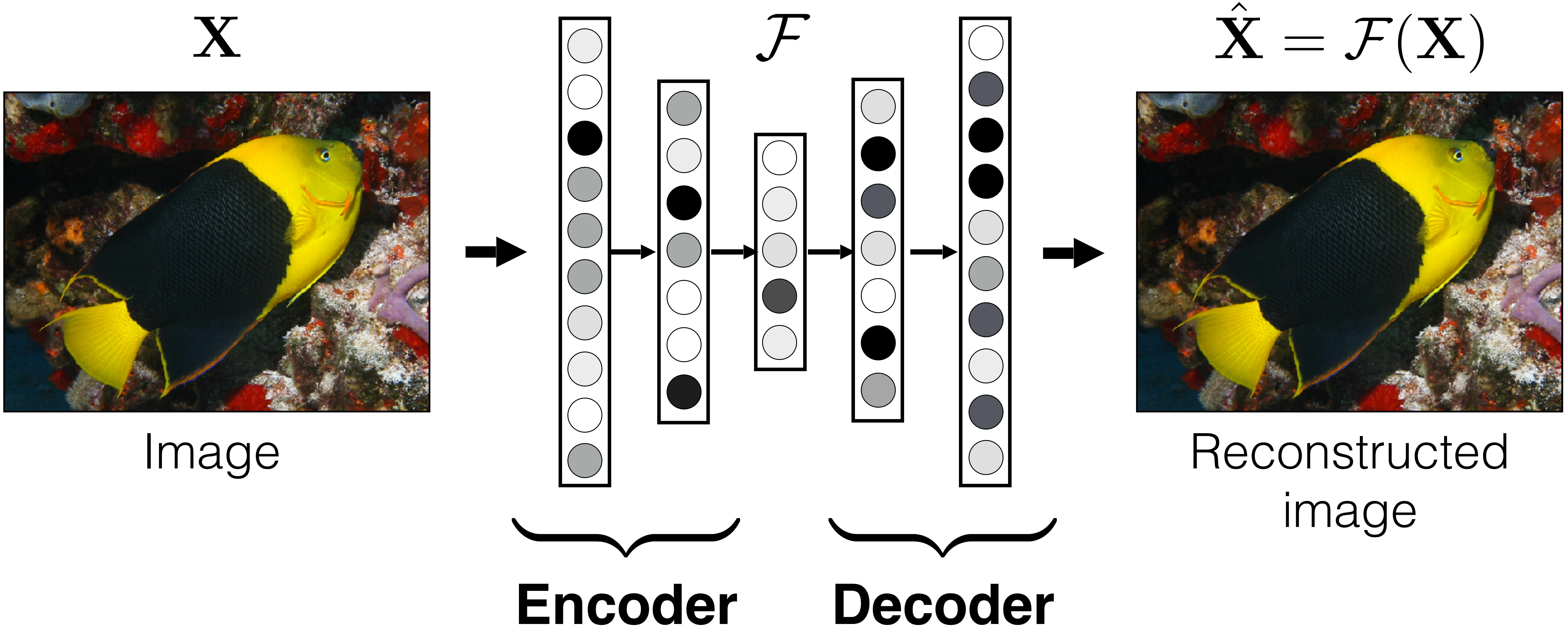
Hypothesis space

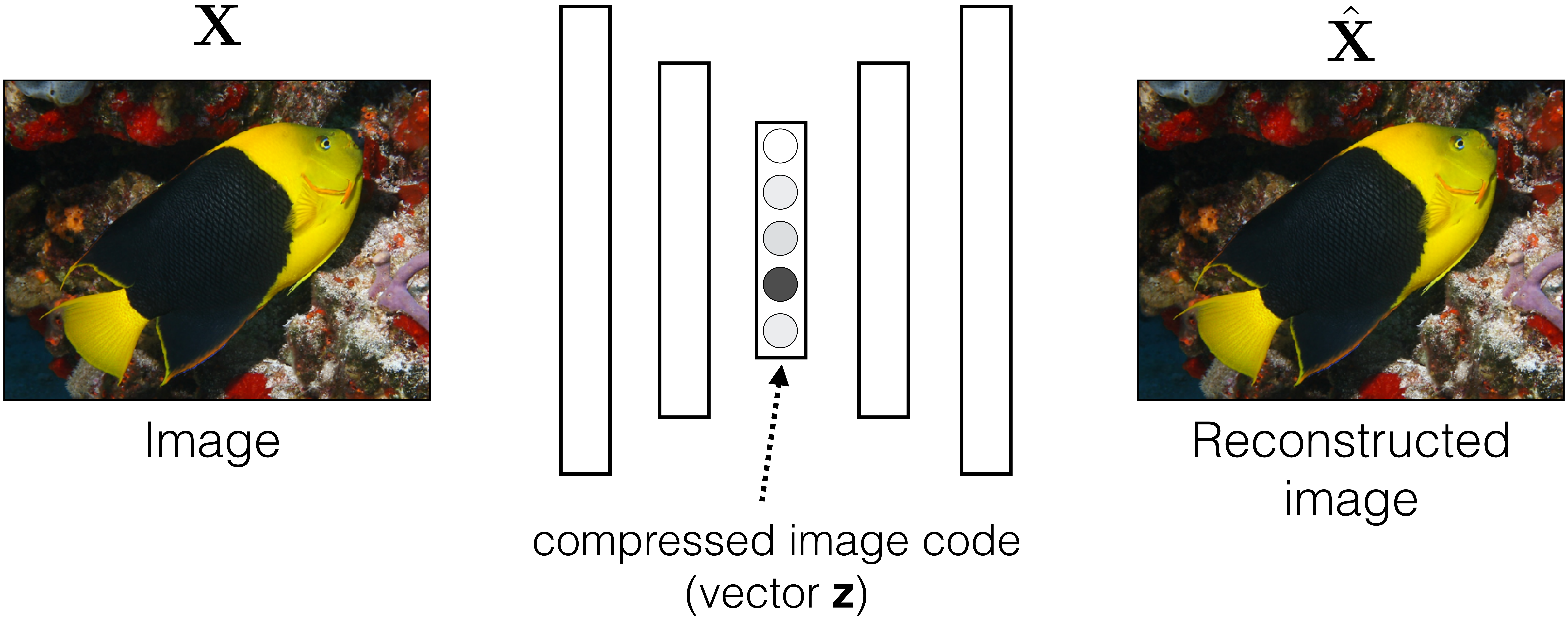
Neural net with a bottleneck

Optimizer

SGD

$\rightarrow f$

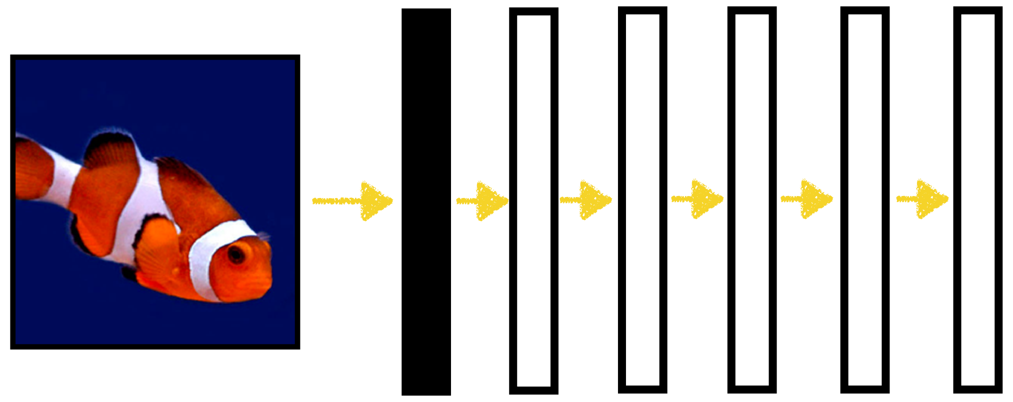

$\mathbf{X}$
Image
$\mathcal{F}$
Encoder
Decoder
$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{X})$
Reconstructed image



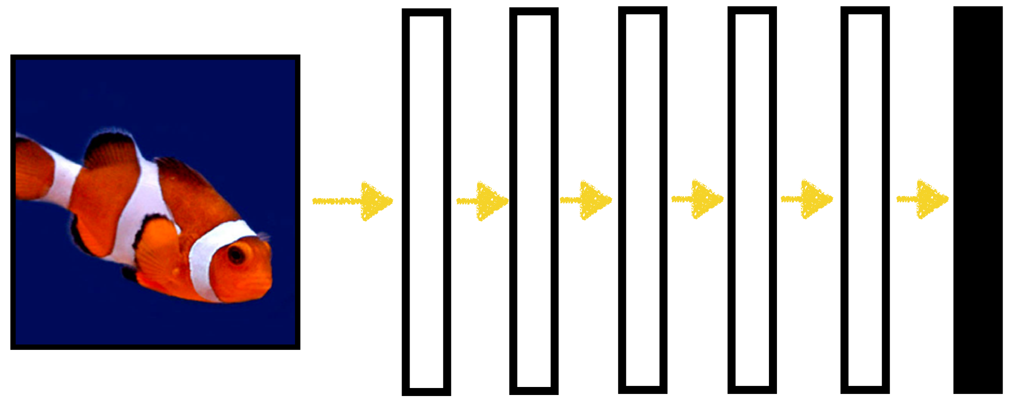
Is the code informative about object class y?

Logistic regression:

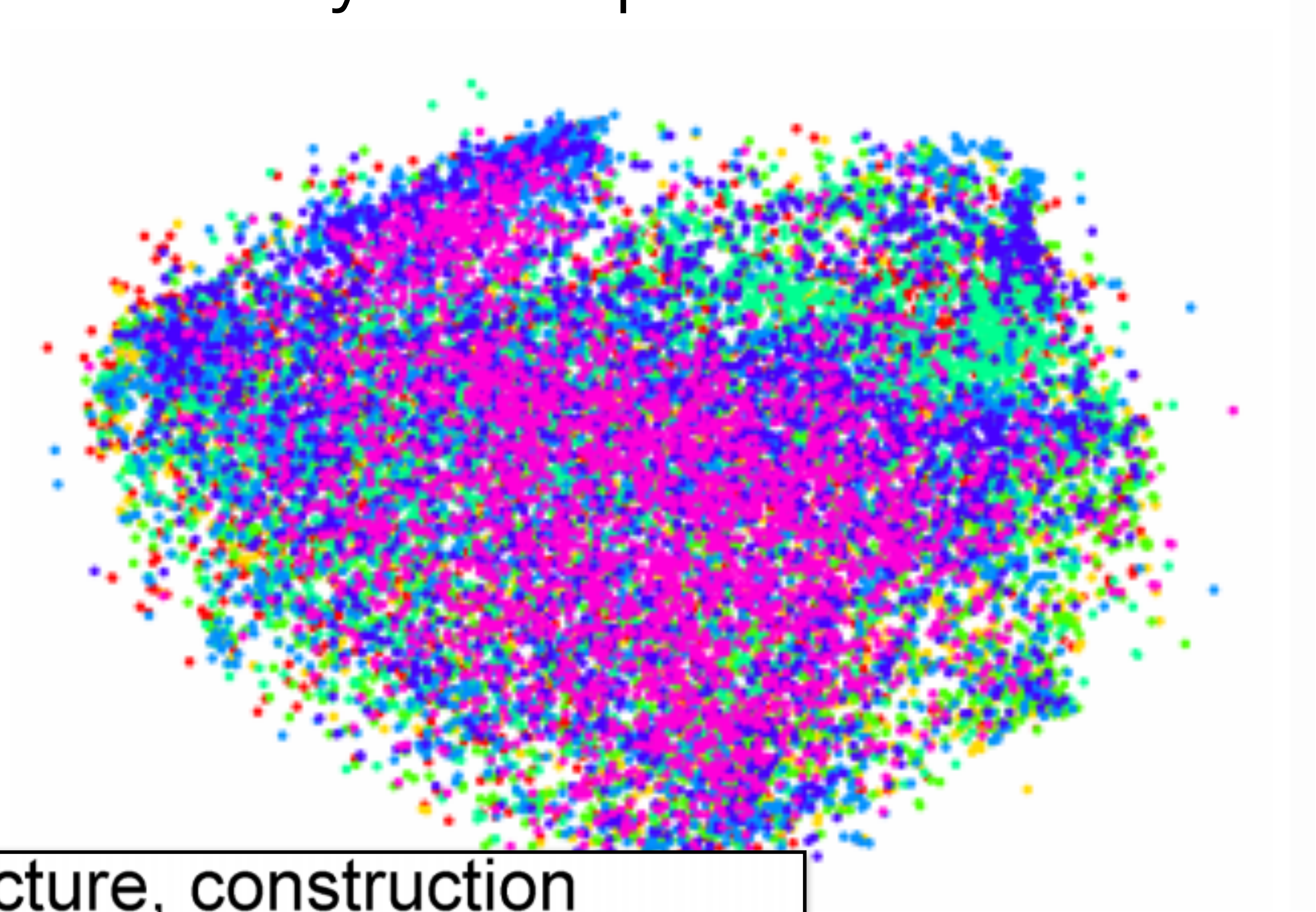
$$y = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b})$$

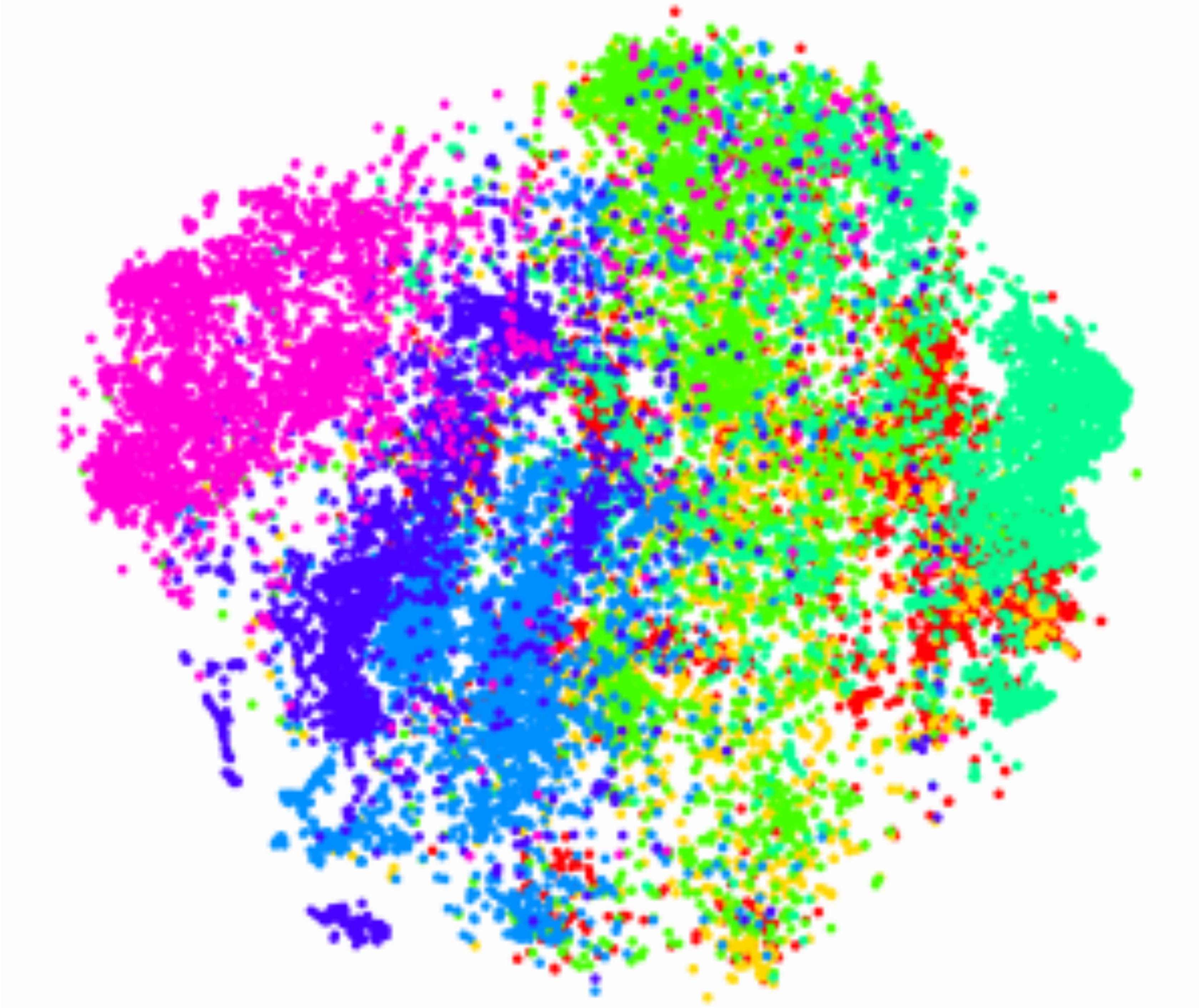


Layer 1 representation



Layer 6 representation



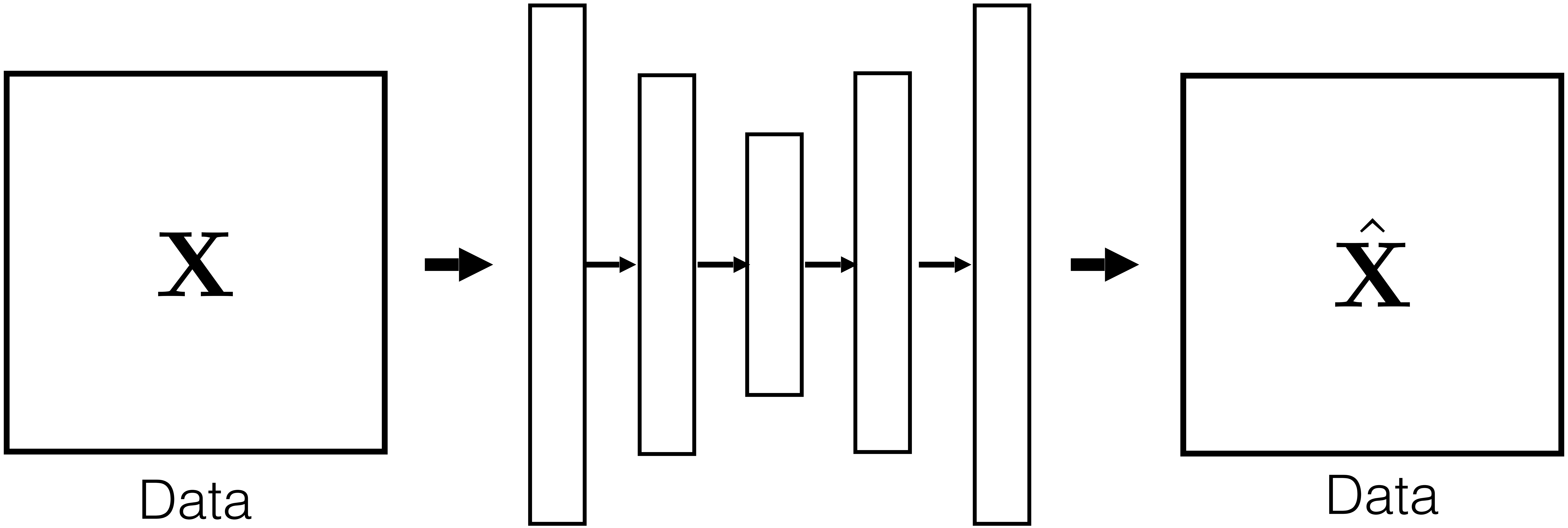


- structure, construction
- covering
- commodity, trade good, good
- conveyance, transport
- invertebrate
- bird
- hunting dog

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

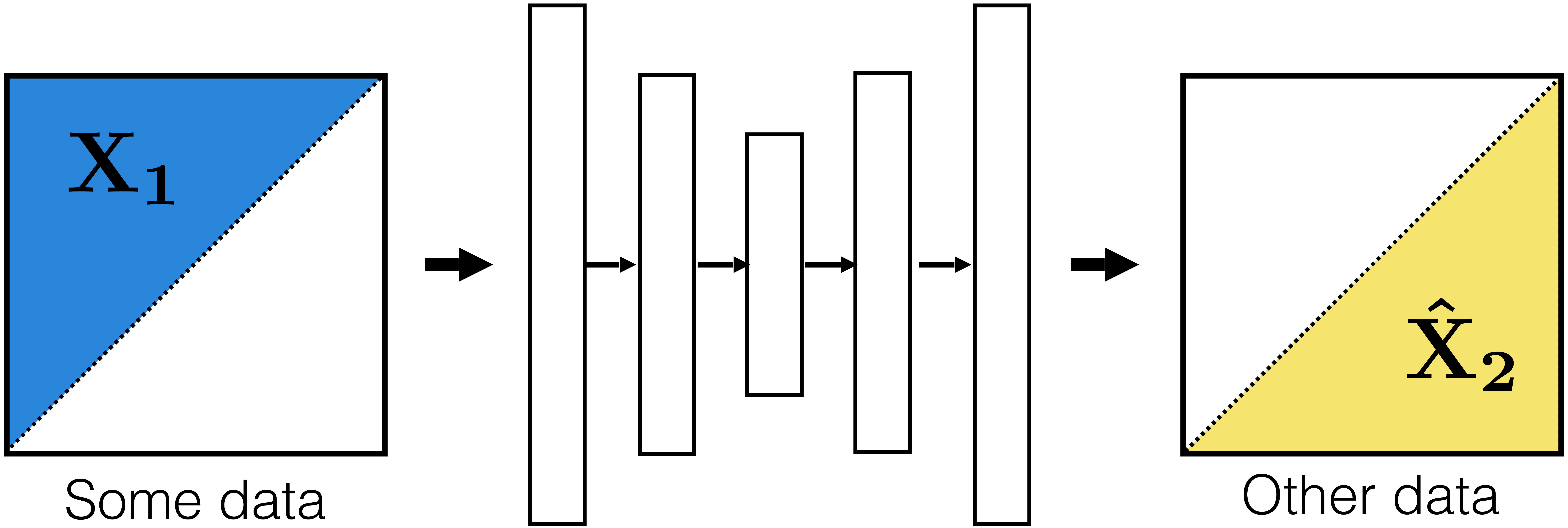
[Visualization technique : t-sne, van der Maaten & Hinton, 2008]

Data compression



Data prediction

aka "self-supervised learning"



Today's Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by
unsupervised learning and **self-supervised** learning?

Why learn representations?

Reinforcement Learning (Cherry)

Predicting a scalar reward given once in a while
A few bits for some samples

Supervised Learning (Chocolate Coat)

Predicting category or vector of scalars per input as provided by human labels.
10-10k bits per sample

Unsupervised / Self-Supervised Learning (Cake)

Predicting parts of observed input or predicting future observations or events
Millions of bits per sample

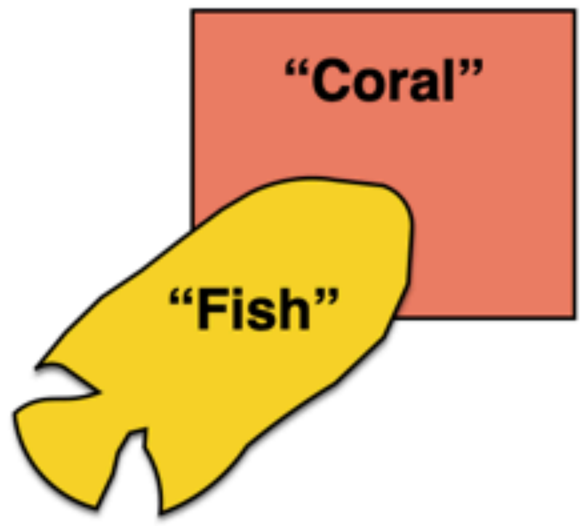
Visualisation Idea by Yann LeCun
Photo by Kristina Paukshtite from Pexels

tl;dr

Representation learning

Good representations are:

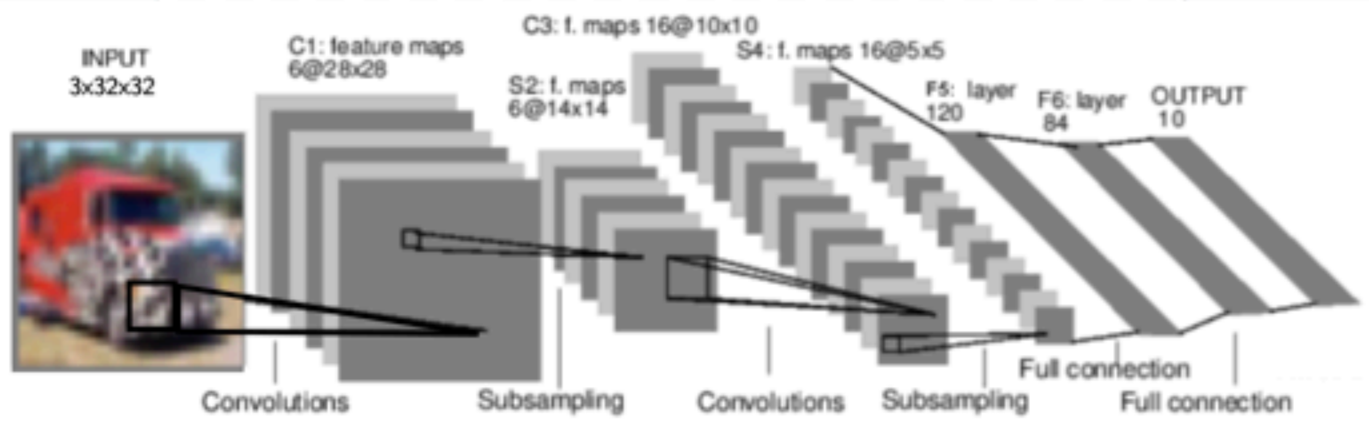
1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Hierarchical (*feature reuse*)
5. *Make subsequent problem solving easy*



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

From MIT 6.8300/6.8301: Advances in Computer Vision

Model Architecture

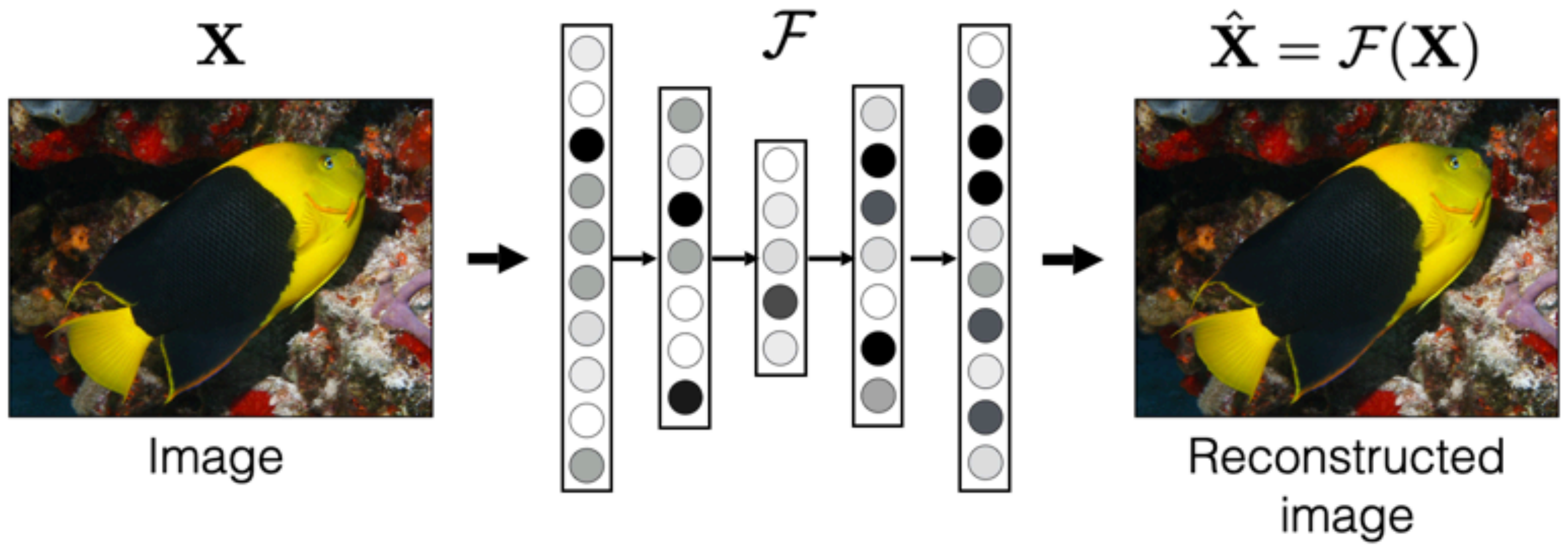


Input:
Image

Output:
Logits for class
(1-10)

classes = ('plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck')

Autoencoder



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

From MIT 6.8300/6.8301: Advances in Computer Vision