Deep Reinforcement Learning

Lecture 5

DNN Backgrounds, Frameworks, and Environments

國立清華大學
NATIONAL TSING HUA UNIVERSITY

National Tsing Hua University
Department of Computer Science

Prof. Chun-Yi Lee

Outline

- Large State MDP Problem
- Case Study — Self-Driving Car & Introduction to Computer Vision
- Deep Learning Frameworks
- Common Reinforcement Learning Environments

Function Approximation

Large state MDP problems

- Large State Reinforcement Learning
  - Shogi : 10^{71} states
  - GO : 10^{170} states
  - Robot Arm : Continuous state space

It is hard to represent all possible states.
Learning value function for each state is slow.

Function Approximation

Large state MDP problems

- Estimate a value function or a policy with function approximation

$$V_\theta(s) = V(s, \theta) \approx V_\pi(s)$$
$$Q_\theta(s, a) = Q(s, a, \theta) \approx Q_\pi(s, a)$$
$$\pi_\theta(a, s) = \pi(a \,|\, s, \theta) \approx \pi(s \,|\, a)$$

- Generalize from seen states to unseen states
- Less memory used in large state MDPs

Function Approximation

Large state MDP problems

Same problem when using images as inputs:

Function Approximation

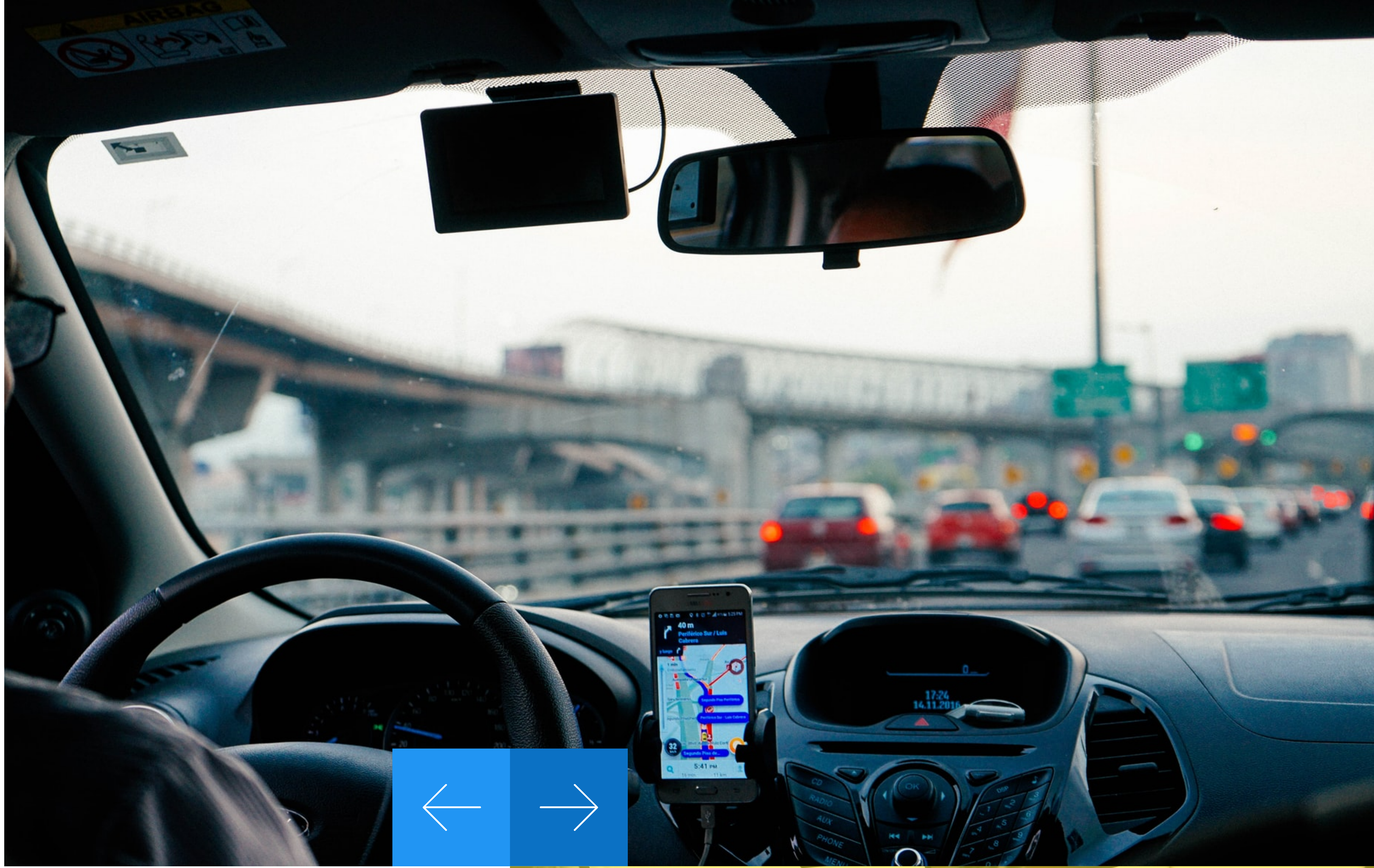
Choose appropriate function approximators

- Many function approximators :
  - Linear Combinations of features
  - Neural networks
  - Decision tree
  - Nearest neighbors
  - Fourier/ wavelet bases

Outline

- Large State MDP Problem
- Case Study — Self-Driving Car & Introduction to Computer Vision
- Deep Learning Frameworks
- Common Reinforcement Learning Environments

Case Study — Self-Driving Car

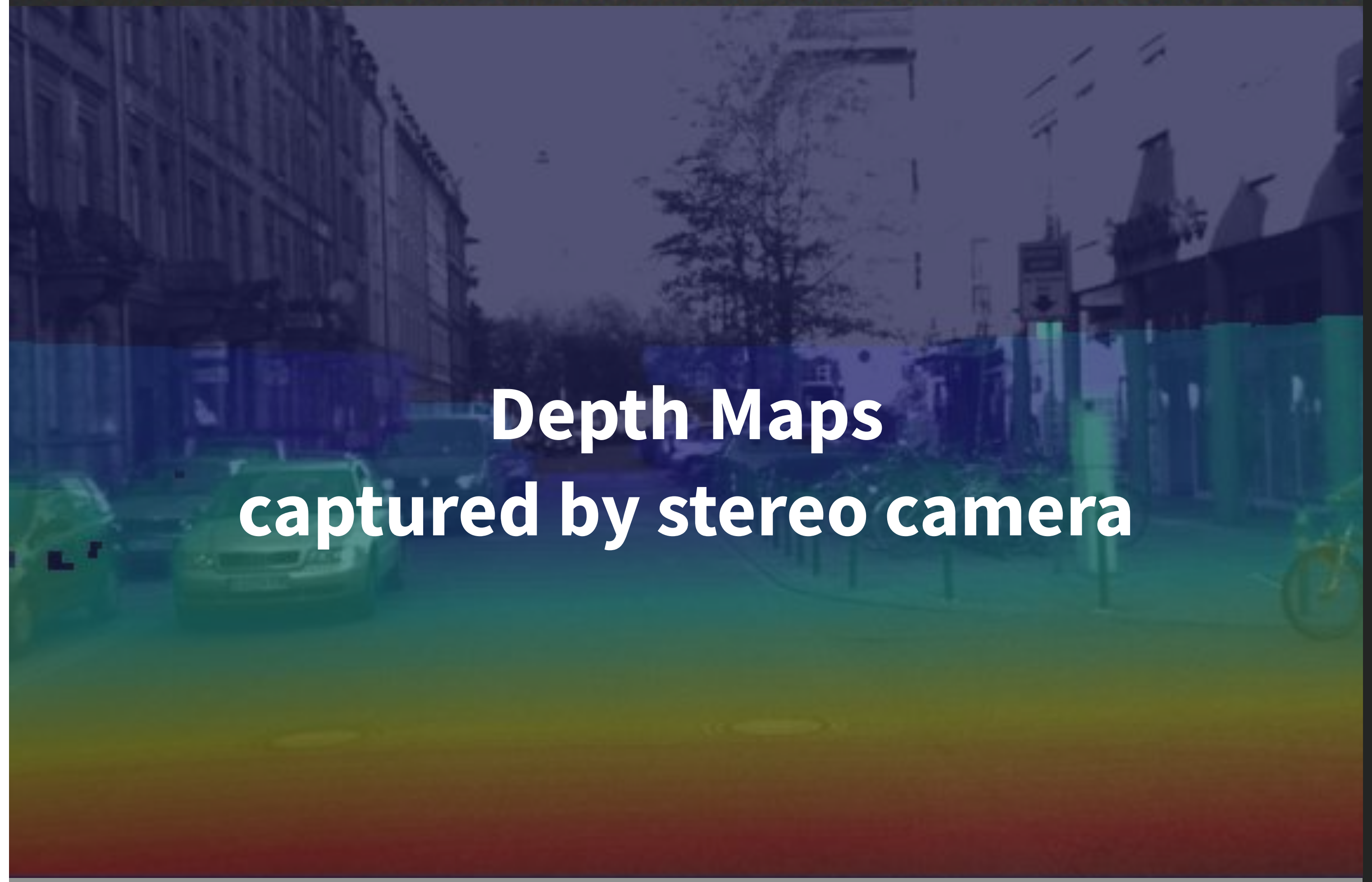


A **self-driving car**, also known as an **autonomous vehicle (AV)**, is a vehicle that is capable of sensing its environment and moving safely with little or no human input.

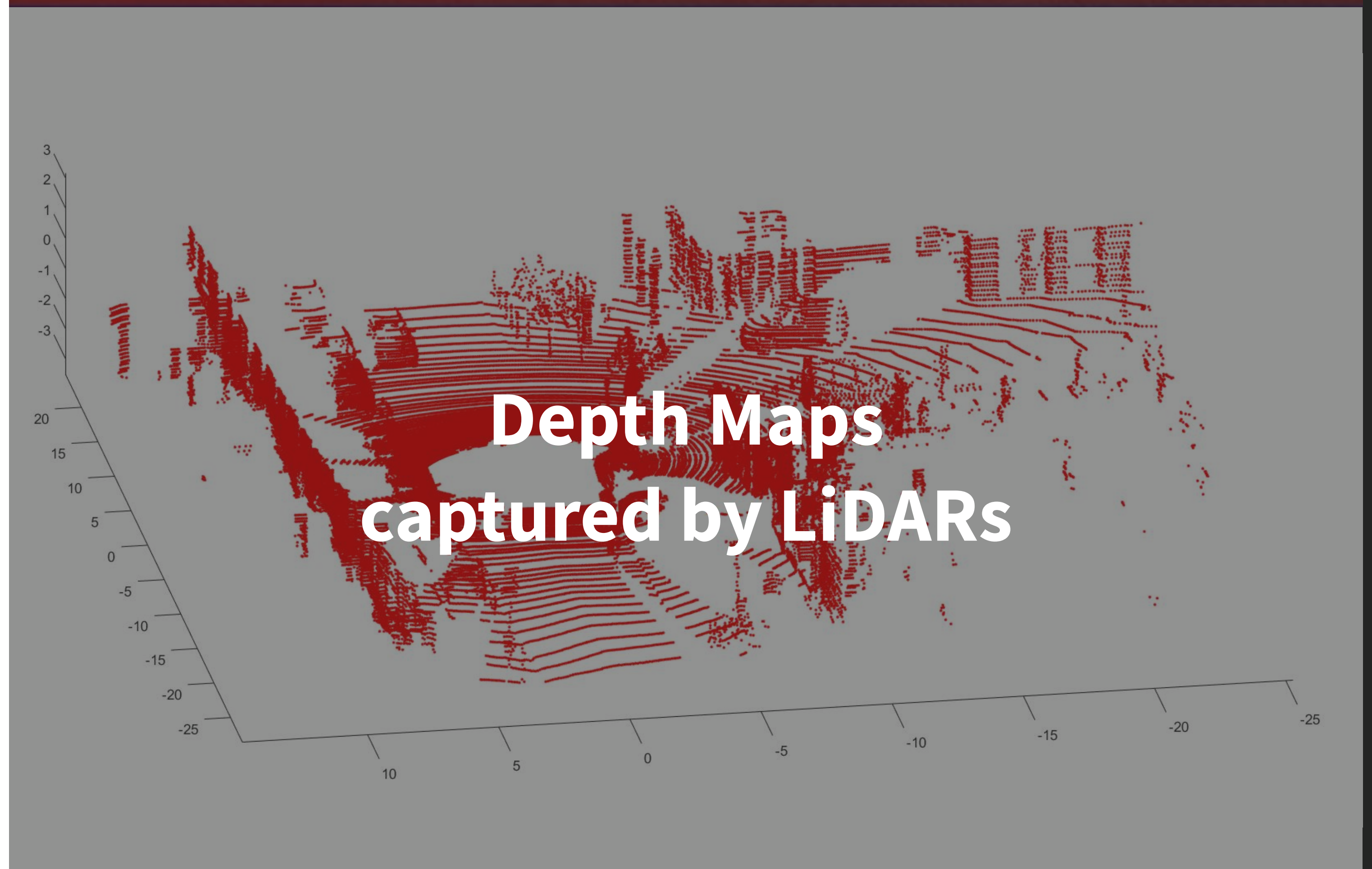
How do Self-Driving Cars See? — Perception

The **perception** of self-driving cars is carried out based on a combination of high-tech sensors and cameras to process and comprehend the environment around the vehicle, in real-time.

- It contains minimally processed data from the camera.
- RGB image is informative, however, it is required to be further processed and/or analyzed.

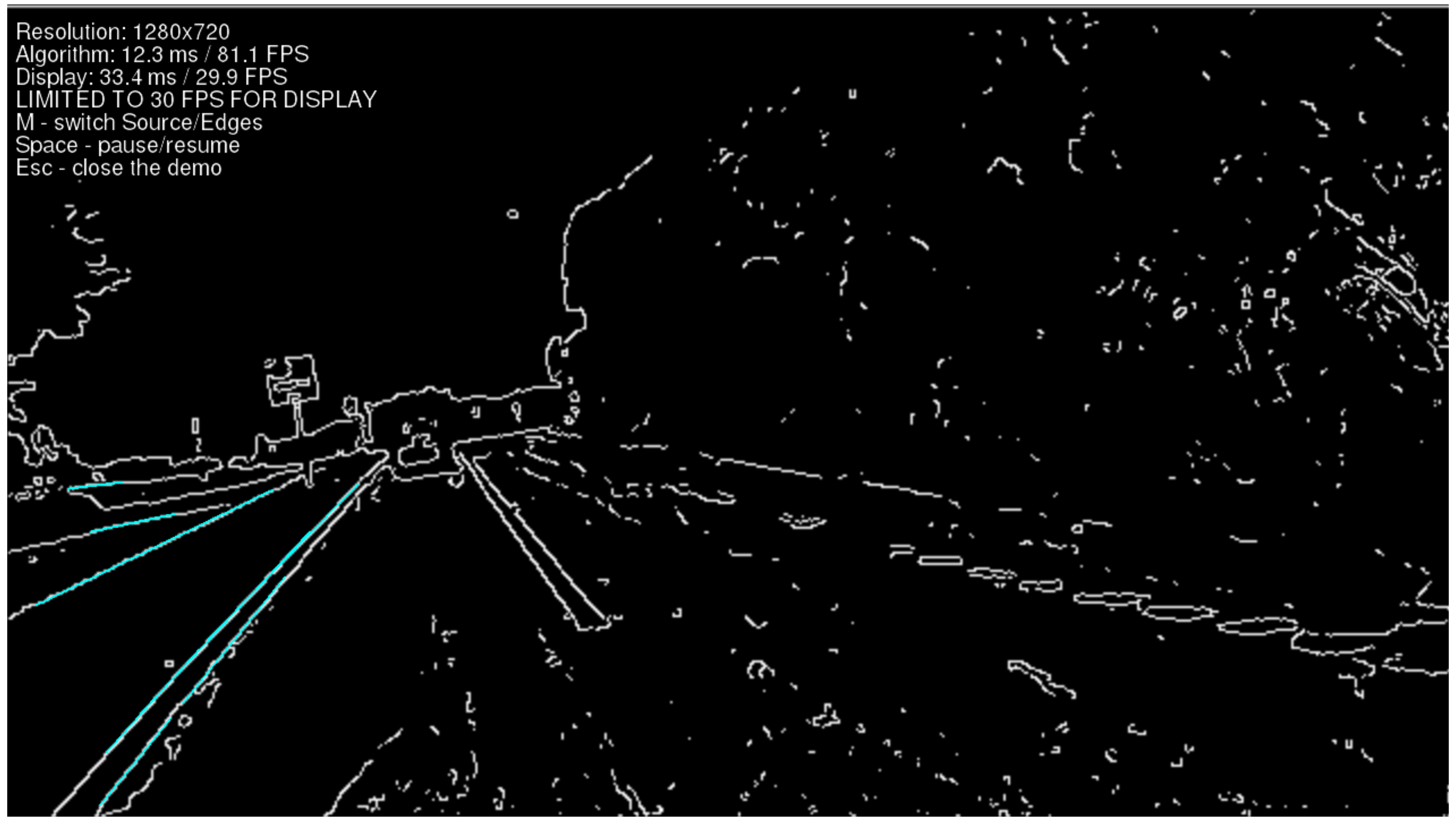


- These cameras simulate a pair of eyes. They provide information on aspects including the position, distance and speed of objects.
- These cameras capture the same scene from two different viewpoints.



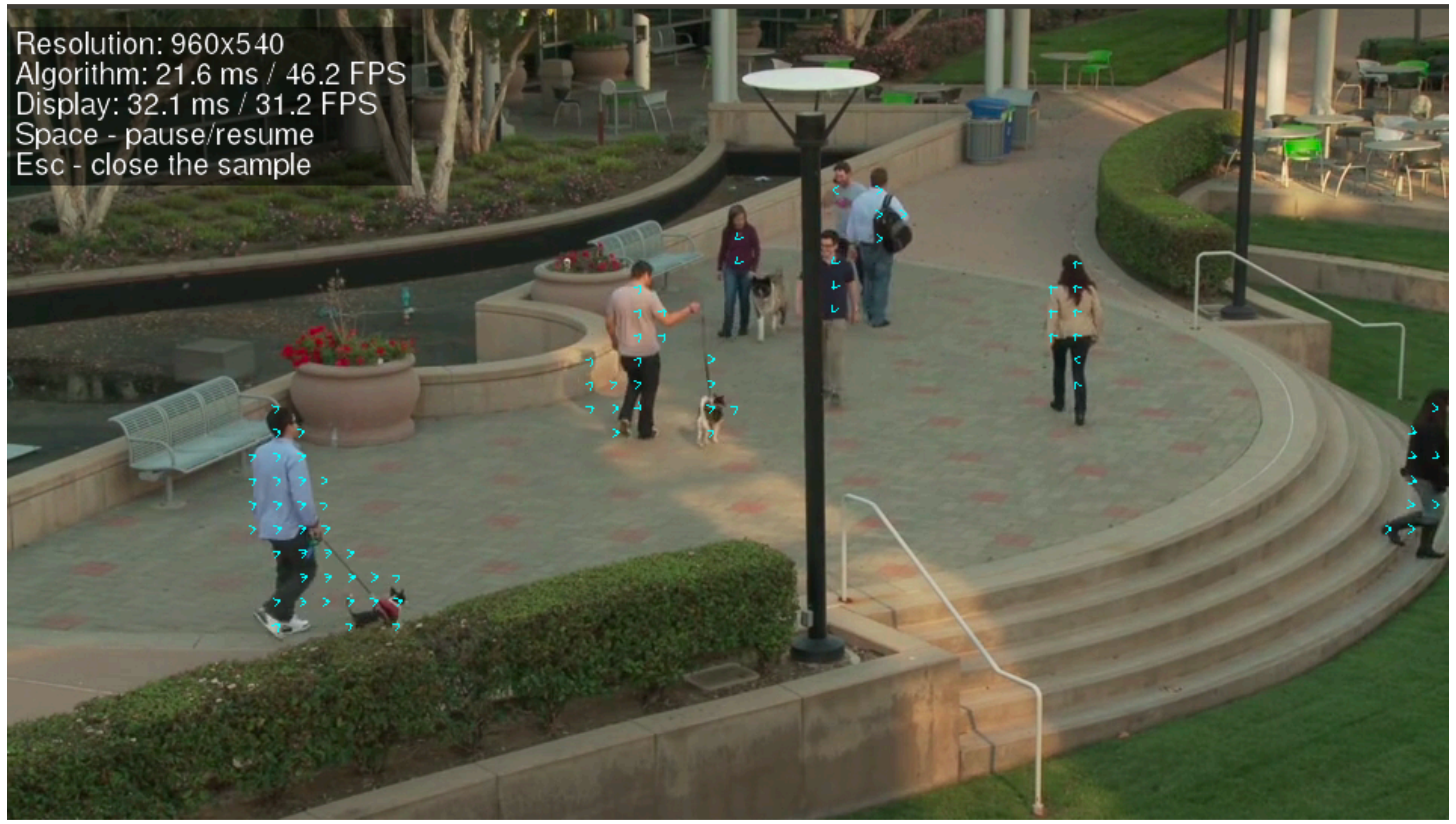
- It provides a 360-degree view of the surrounding.
- LiDAR system sends thousands of laser pulses every second.
- The pulses collide with the surrounding objects and reflect back. The resultant light reflections are then used to generated a 3D point cloud.

Conventional Ways of Processing Raw RGB Images



Hough line transform

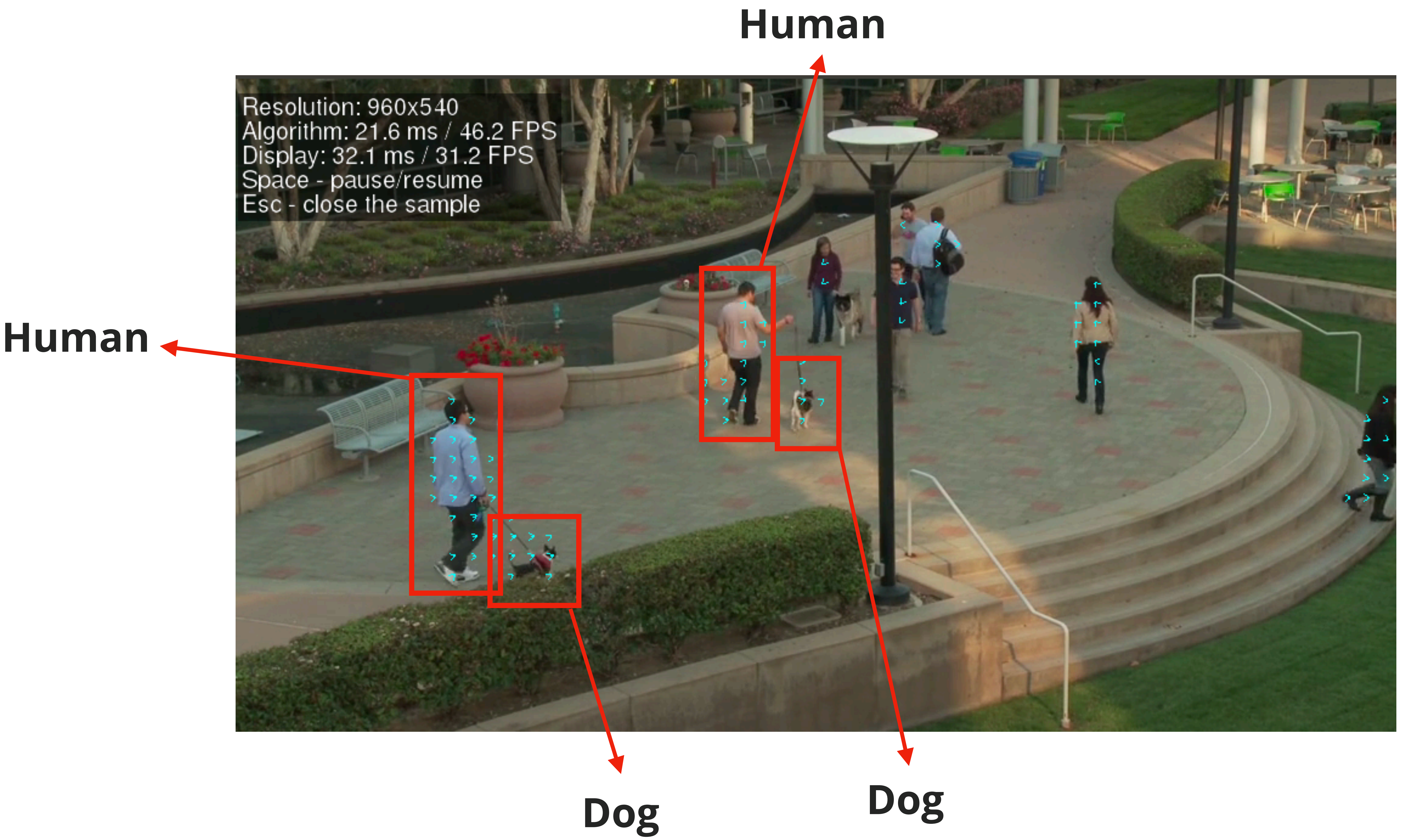
It is a popular technique to **detect any shape**, if you can represent that shape in mathematical form. **It can detect the shape even if it is broken or distorted** a little bit.



Motion estimation

Motion estimation **examines the movement of objects** in an image sequence to try to obtain **displacement vectors**, which plays an important role in **crowd analytics and behavior recognition**.

How to Recognize Important Things?



Features for Recognizing Images

Manually Defined Features

Features are defined by heuristics or hard-coded formula, not adaptable to new or different objectives.

Learning-based Features

Features are learned based on the data presented to learning-based models, such as deep neural networks.

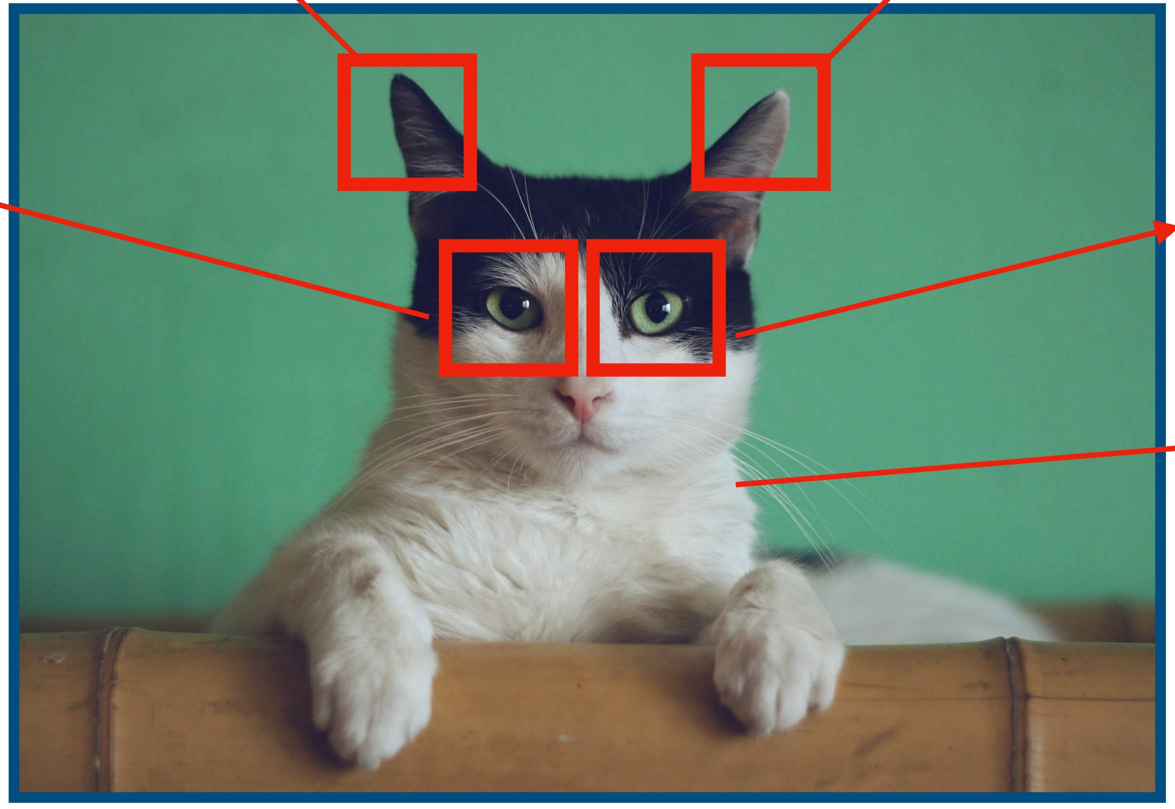
Features for Recognizing Images

Manually Defined Features

Triangle-like shape /
In the top of image

Triangle-like shape /
In the top of image

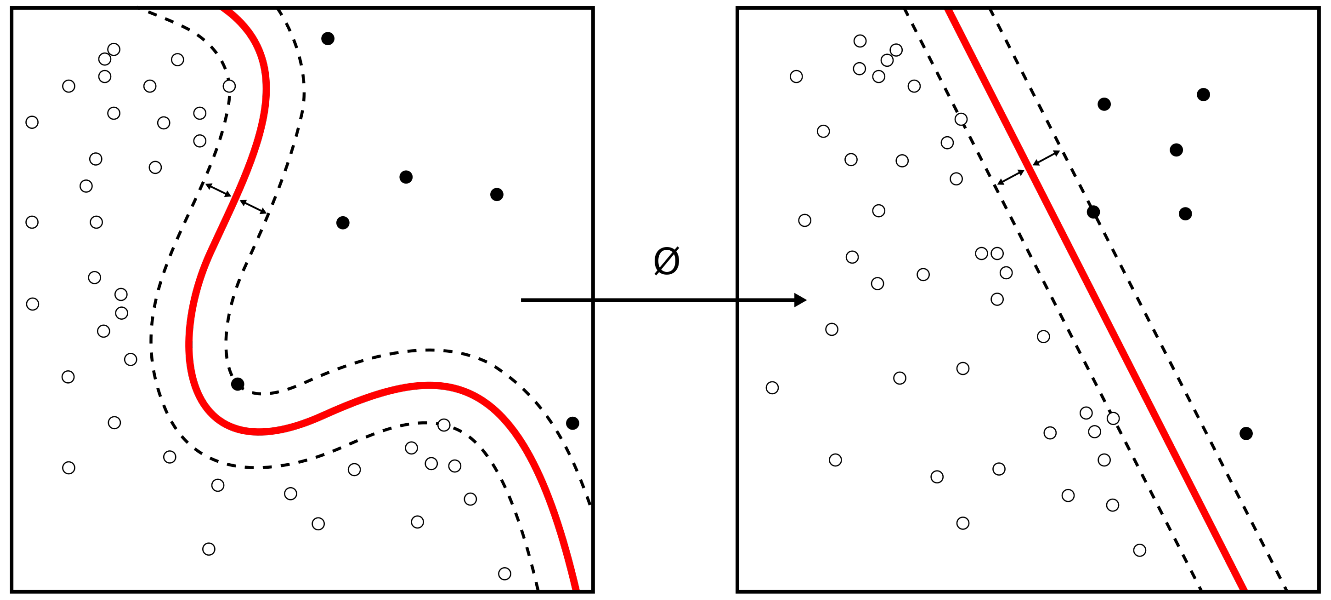
Circle-like shape /
In the middle of image



Circle-like shape /
In the middle of image

Fluffy skin / White

Manually Defined Features



Traditional Machine Learning
Classification Approaches, e.g., SVM.

Features for Recognizing Images

Occlusion

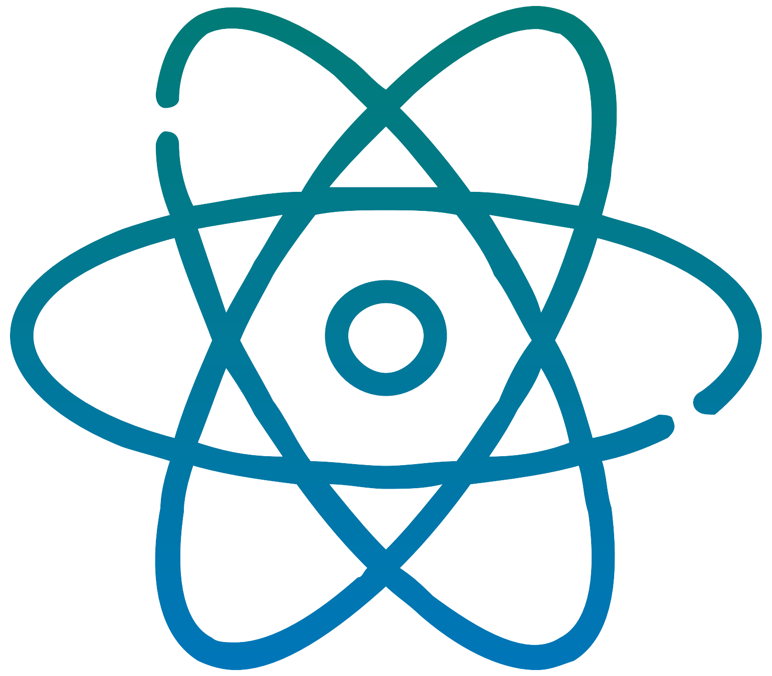
Background clutter

Illumination

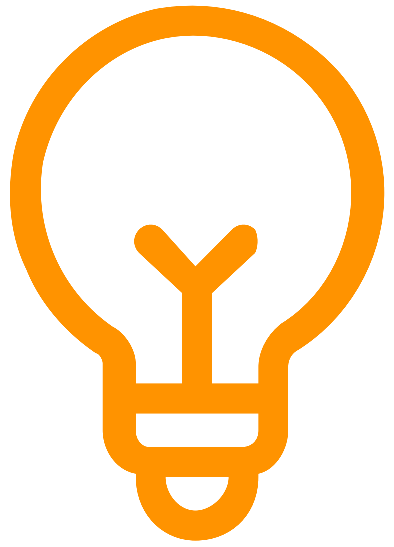
There is no simple way to manually define features to represent a cat.
This is because an image can be represented in various perspectives.

Features for Recognizing Images

A lot of cat and dog images



Learn from experience
(trial and error)



Find good features
=>
Know how to
classify cats and dogs

Visualize and Explain the Learned Features!

Low-level features **High-level features**

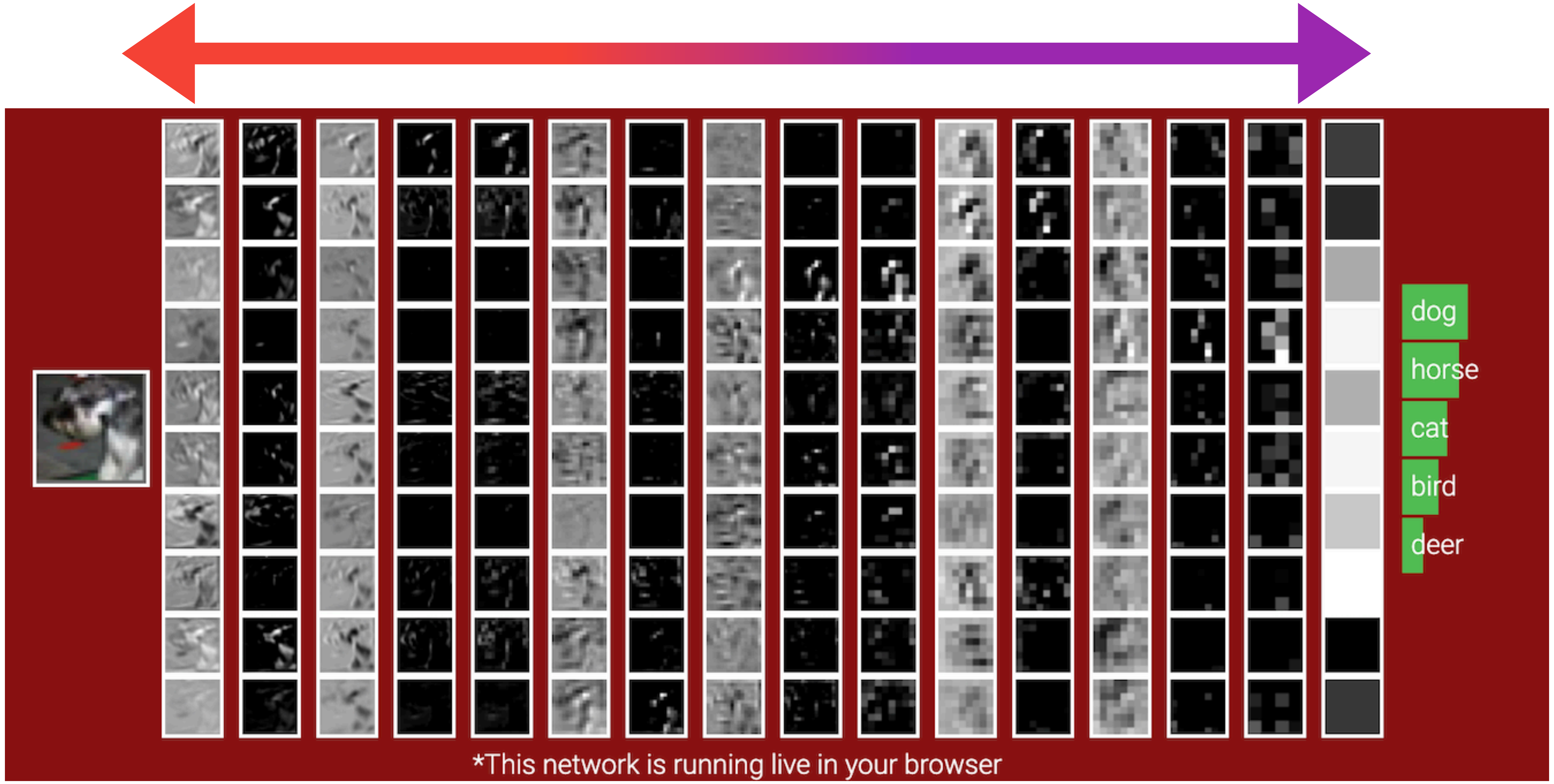


Image from CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2019.

Different Applications of Computer Vision

SINGLE OBJECT

CAT

Classification

There is a cat in the image.

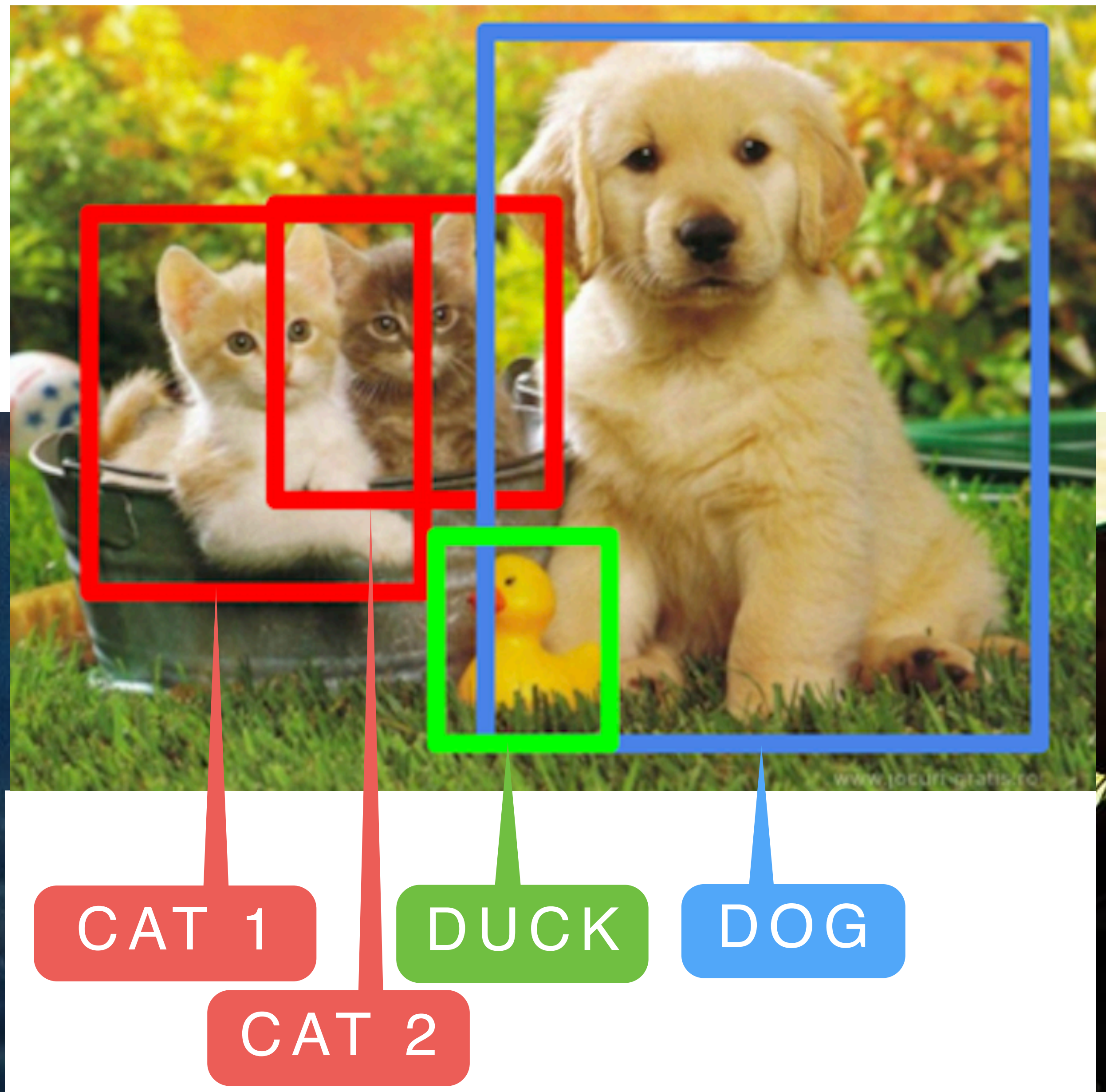
CAT

Classification + Localization

There is a cat at the center of the image.

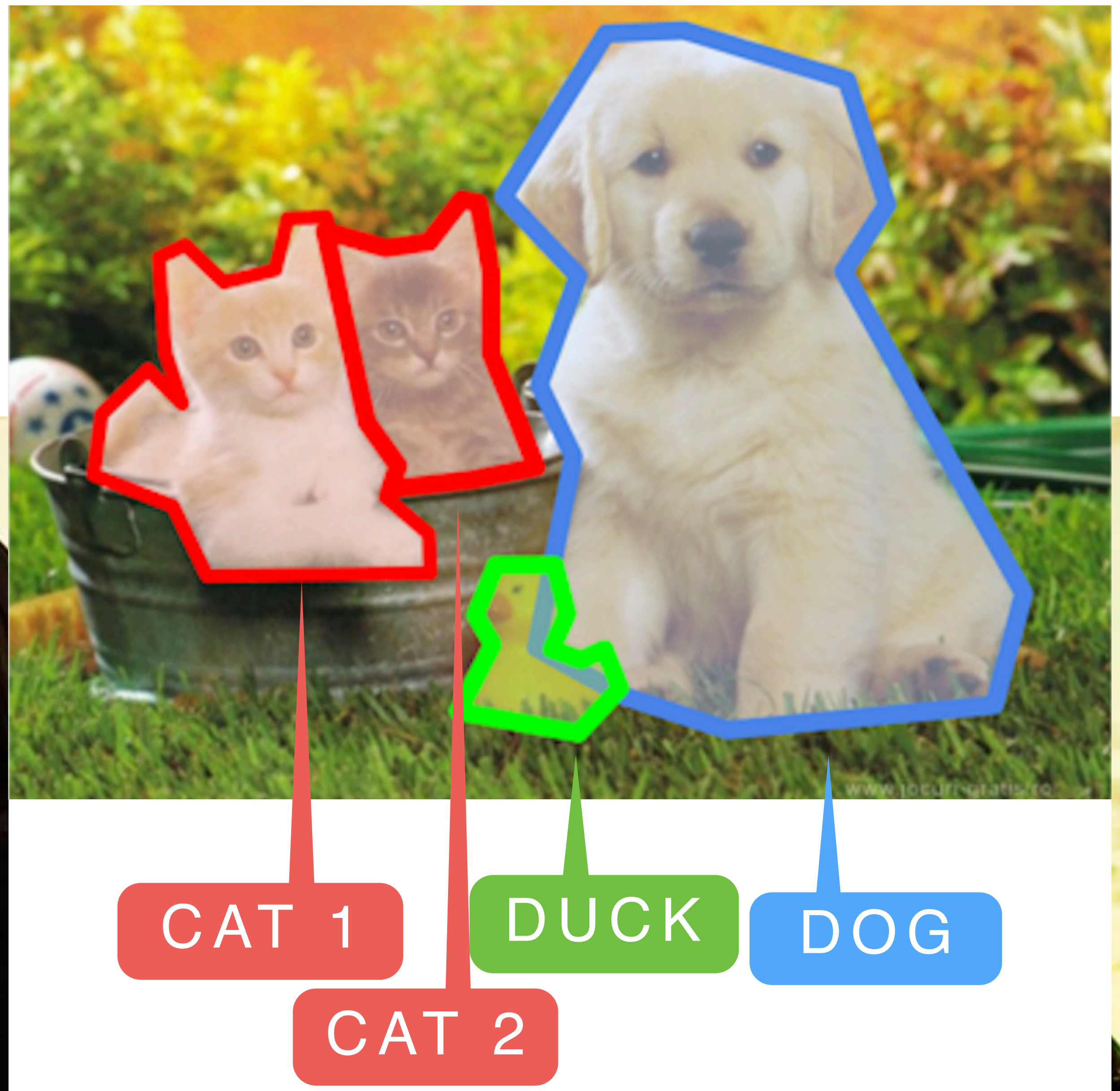
Different Applications of Computer Vision

MULTIPLE OBJECTS



Object Detection

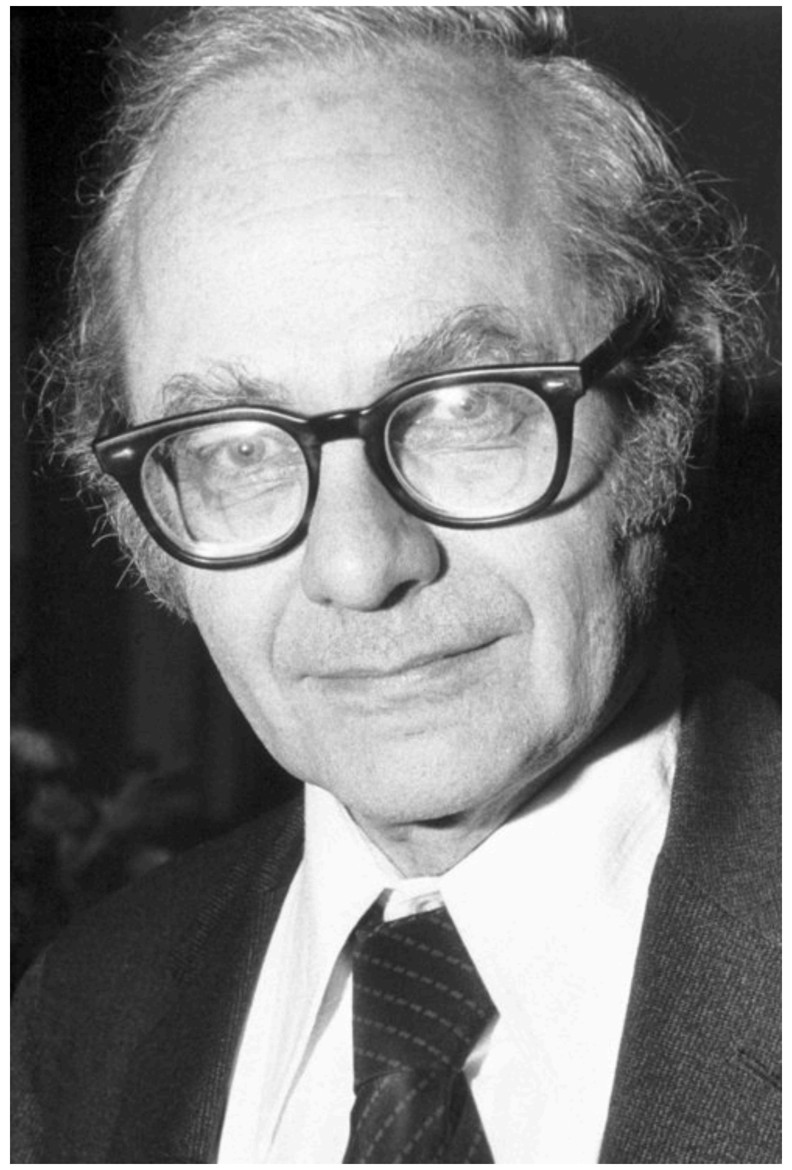
There are **two cats (left), one duck (center), and one dog (right)** in the image surrounded by the **bounding boxes**.



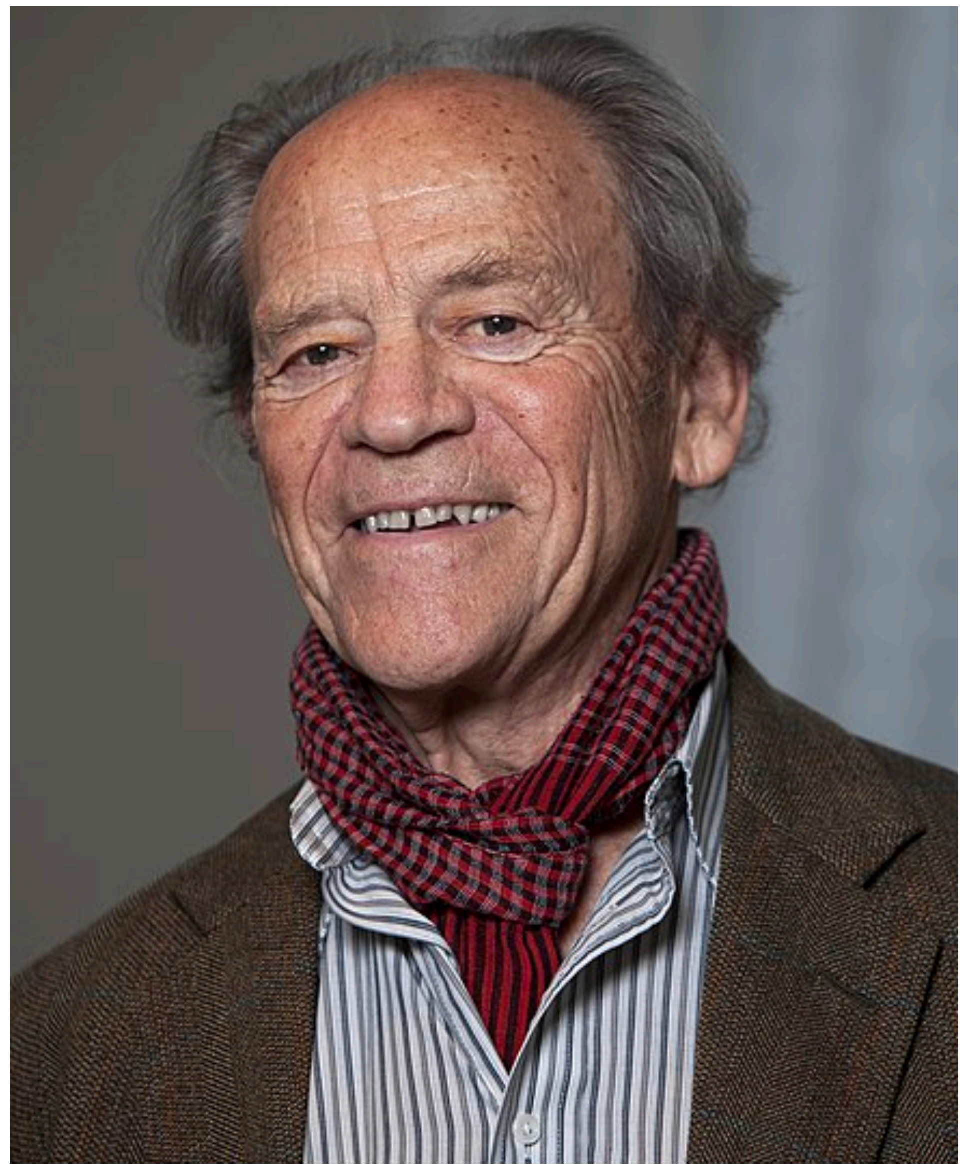
Instance Segmentation

There are **two cats (left), one duck (center), and one dog (right)** in the image highlighted by the **contours**.

History of Computer Vision — Inspired by Neuroscience



David H. Hubel



Torsten Wiesel

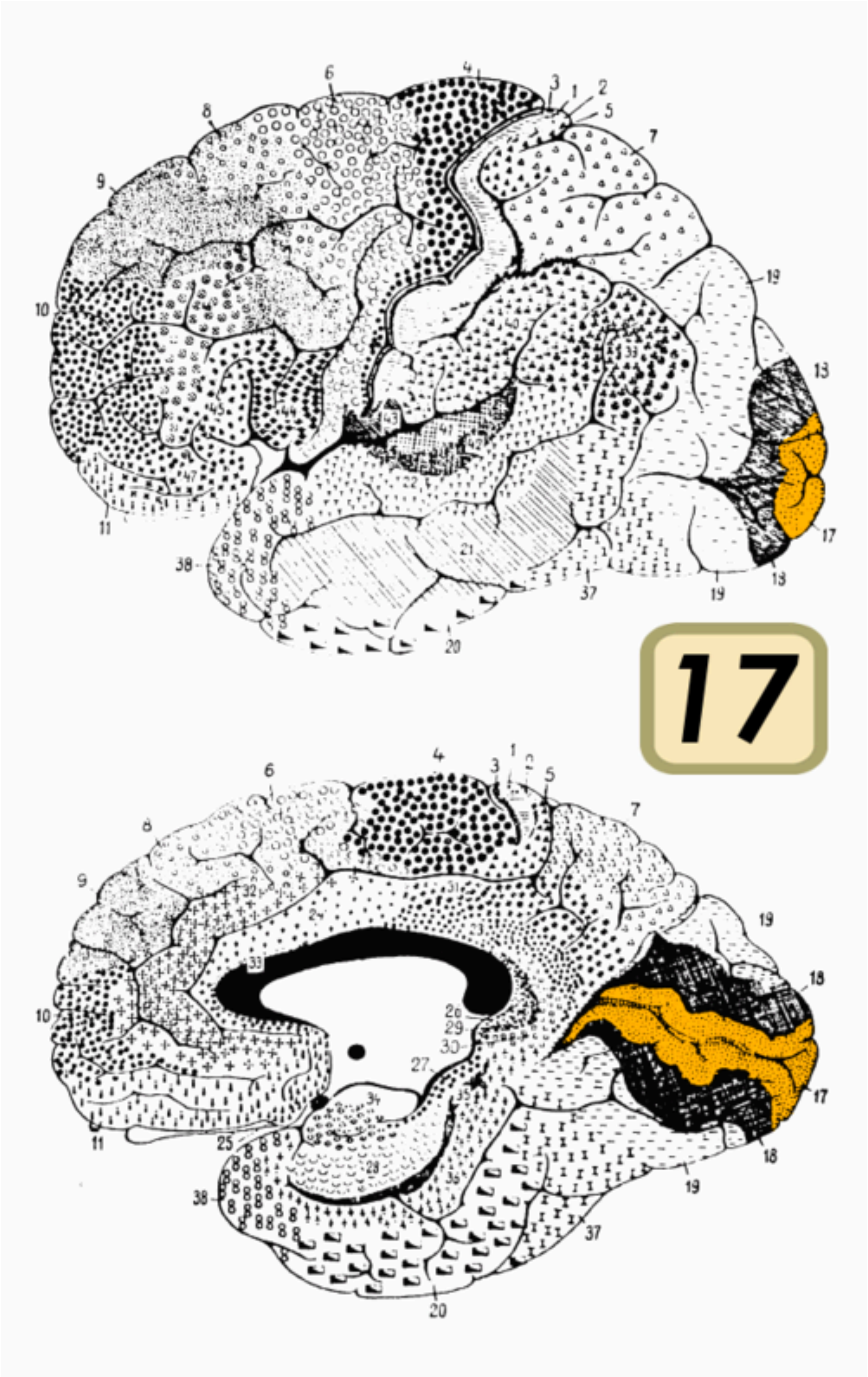
Received the 1981 Nobel Prize in Physiology or Medicine

Images from "Wikipedia".

The research of computer vision can be dated back to researches in biological experiments in 1950s.

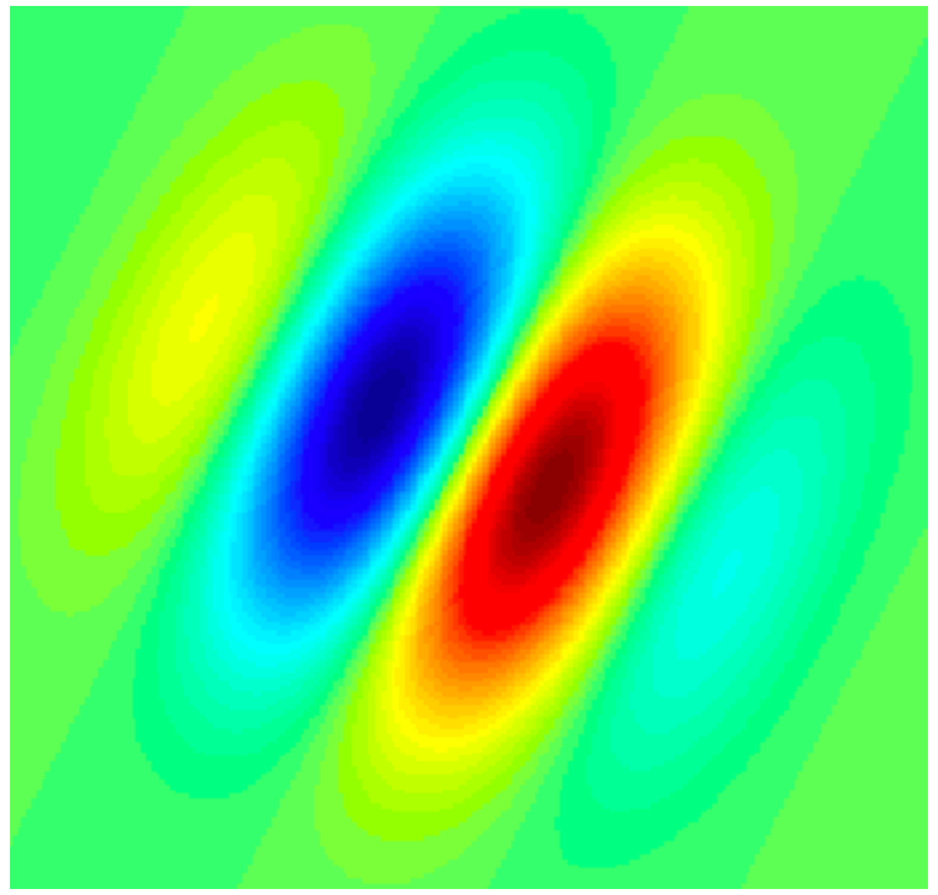
- They described **"simple cells"** and **"complex cells"** in human visual cortex.
- Both the cells are used in **pattern recognition**.

History of Computer Vision



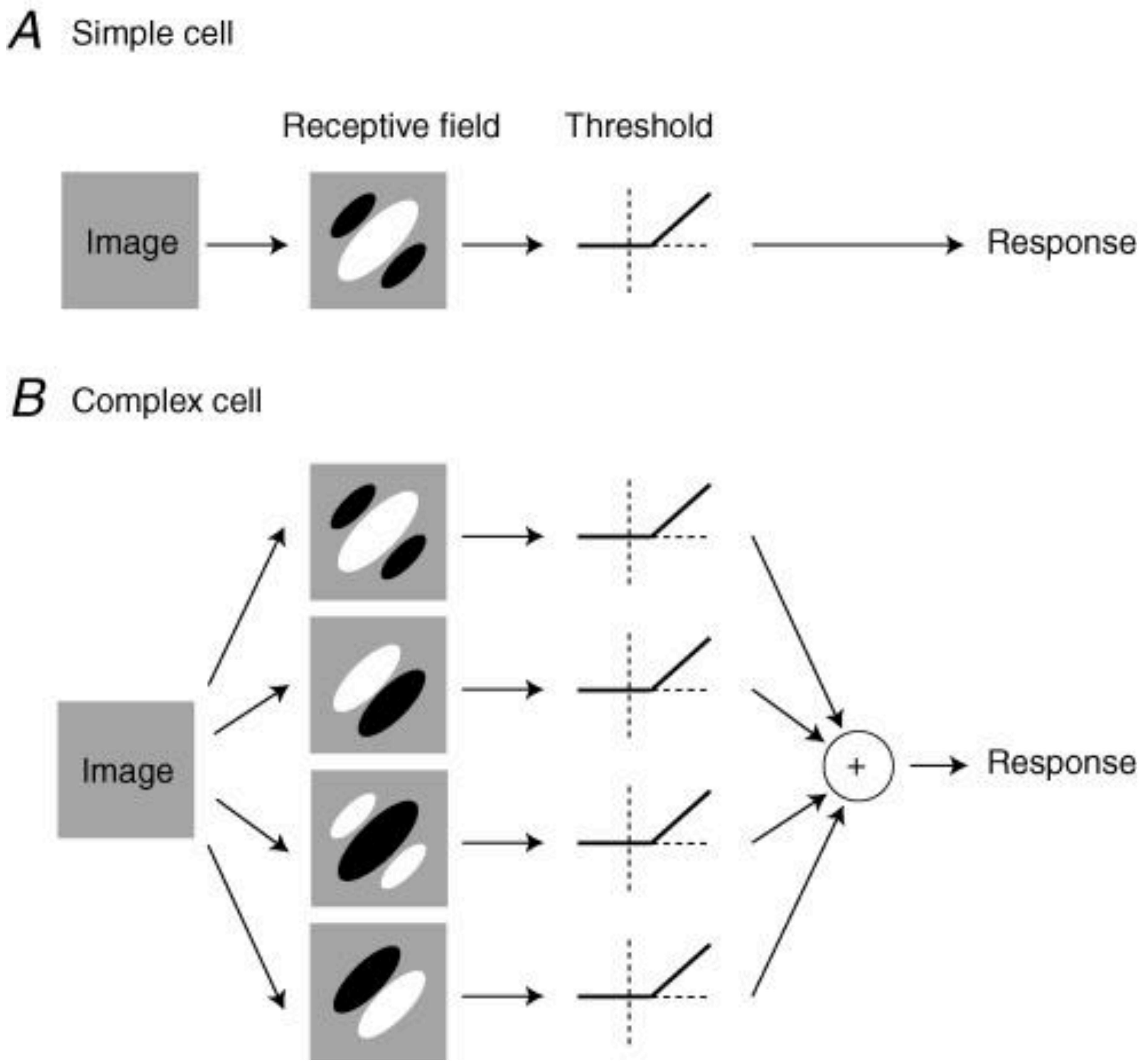
Primary Visual Cortex

- A simple cell **responds to edges and bars** of **particular orientations**.
- A complex cell responds primarily to oriented edges and bars like simple cells, however, it has a degree of **spatial invariance**.



Receptive Fields

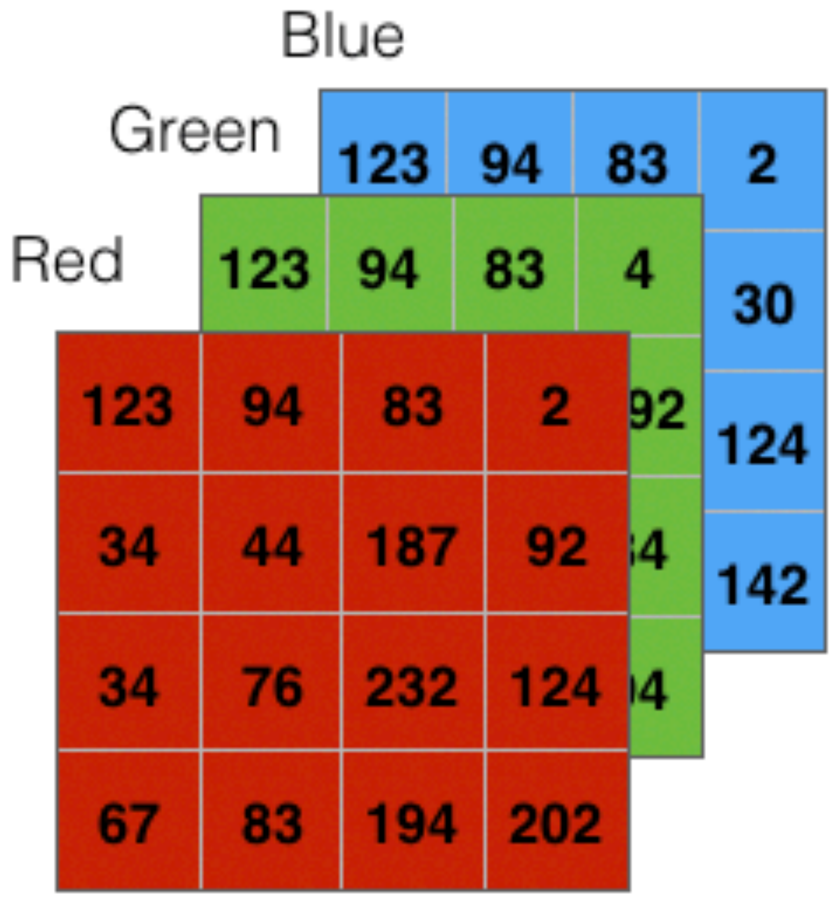
History of Computer Vision



The models of simple and complex cells proposed by Movshon *et al.* 1978*a*,*b*

What is Convolution? And Why?

=



Human's perception

What computers see

What is Convolution? And Why?

撫子

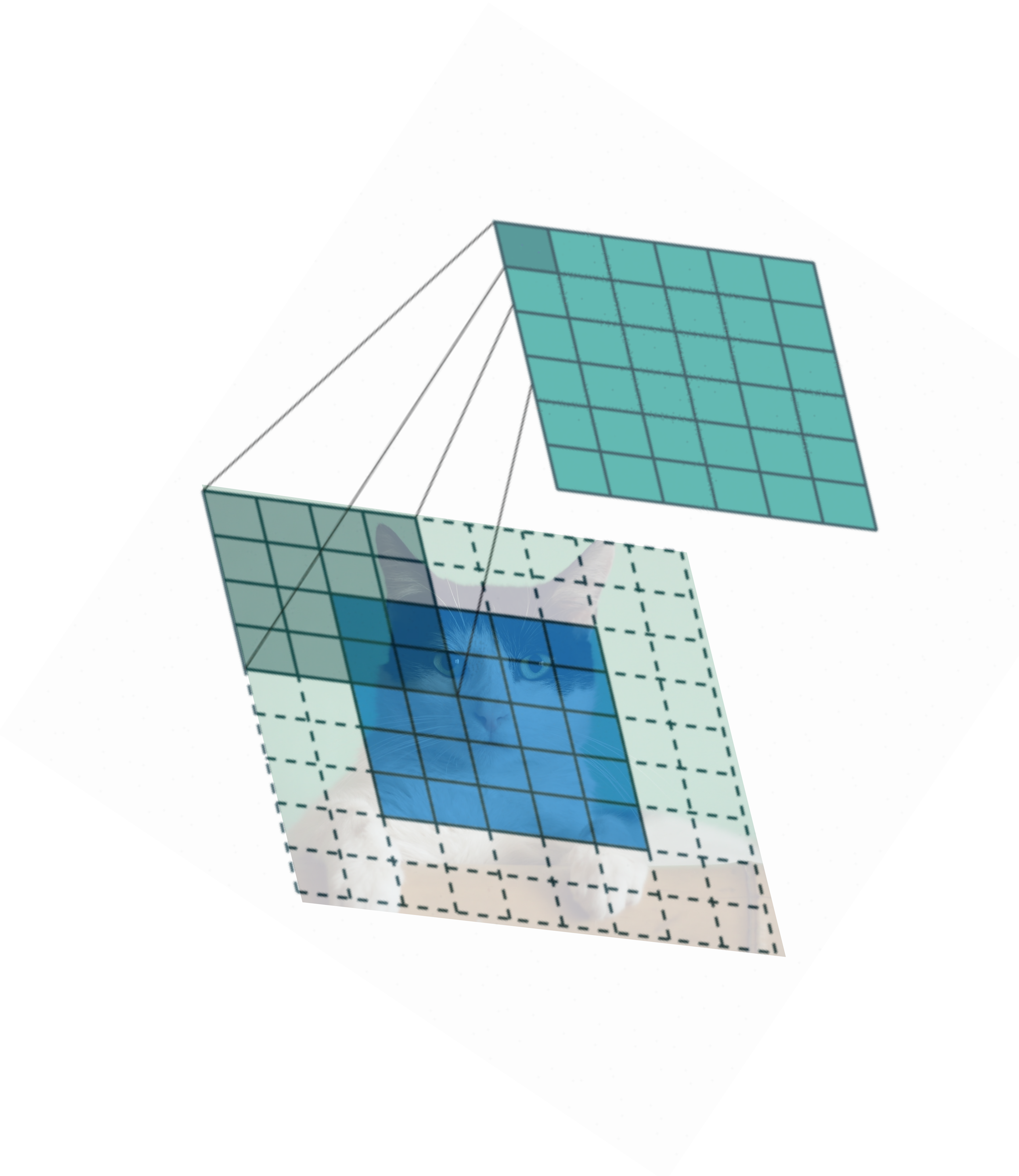
Also 撫子

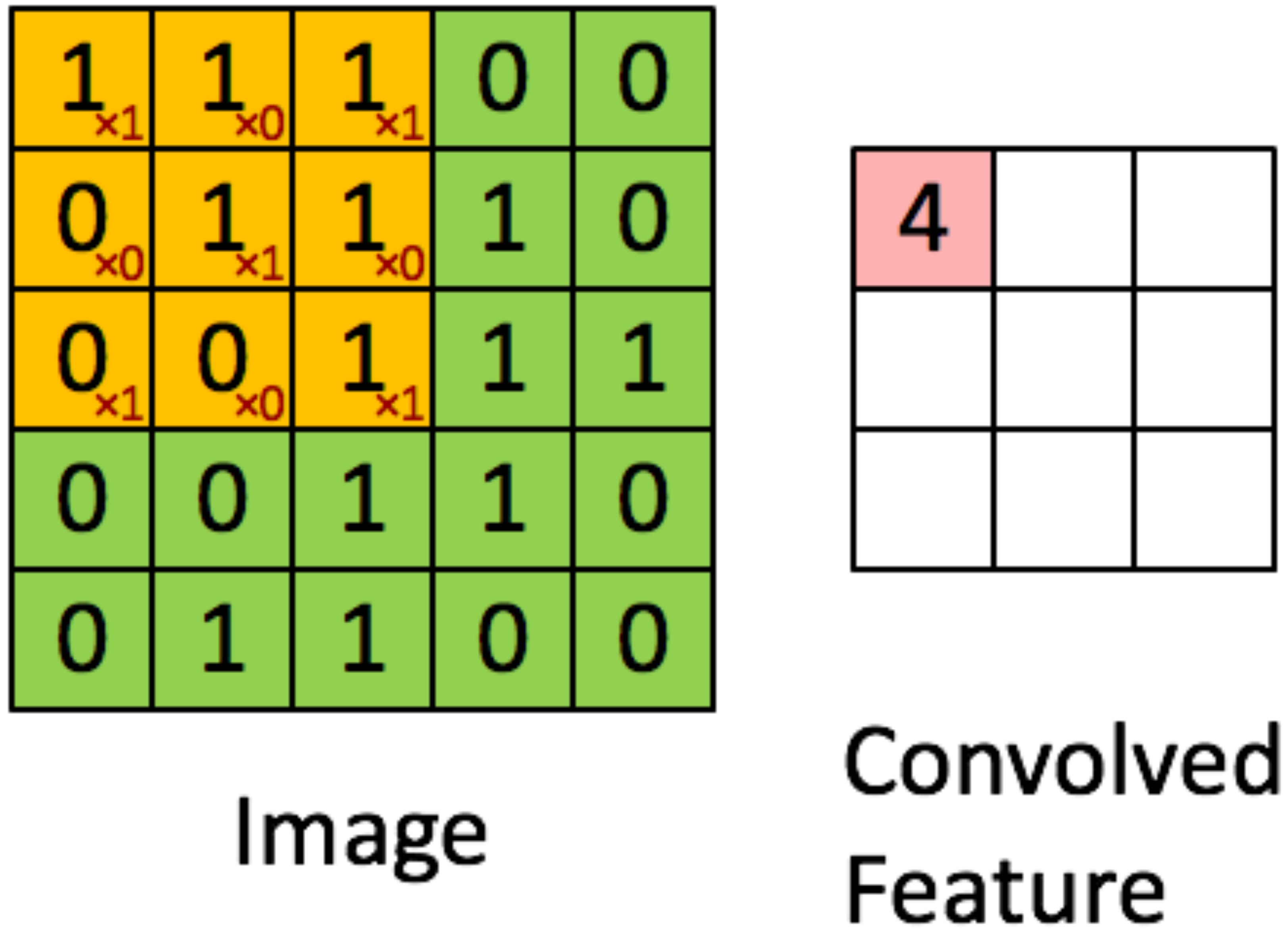
Still 撫子

Spatial Invariance

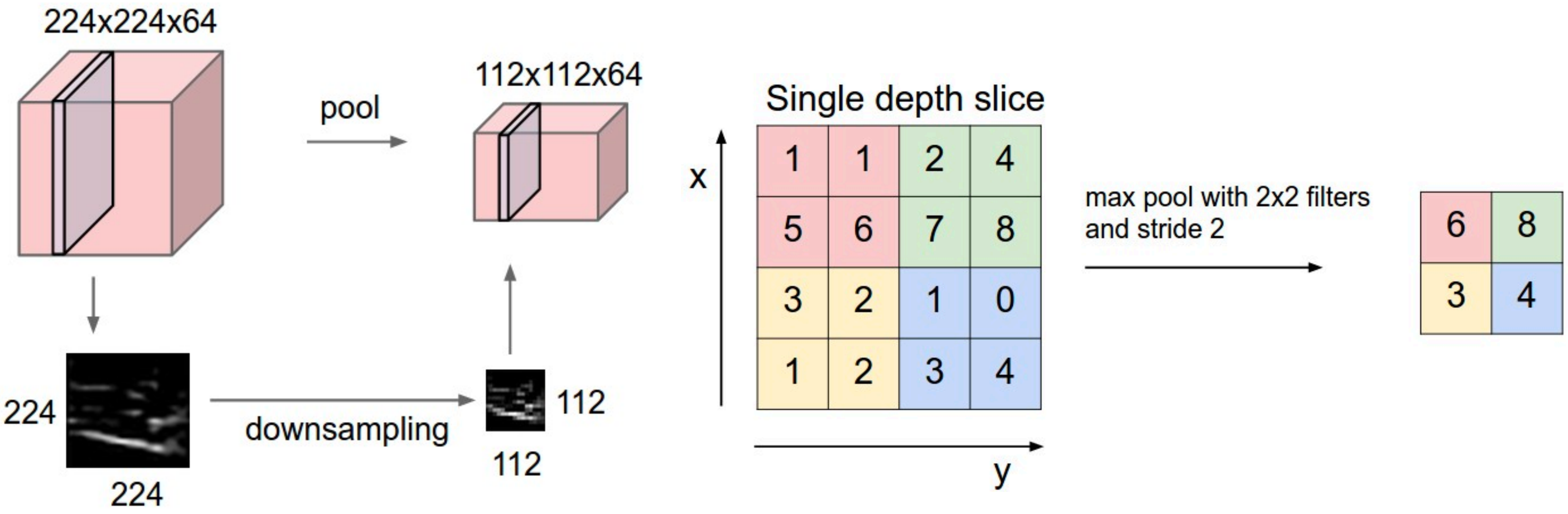
Local Dependence

What is Convolution? And Why?



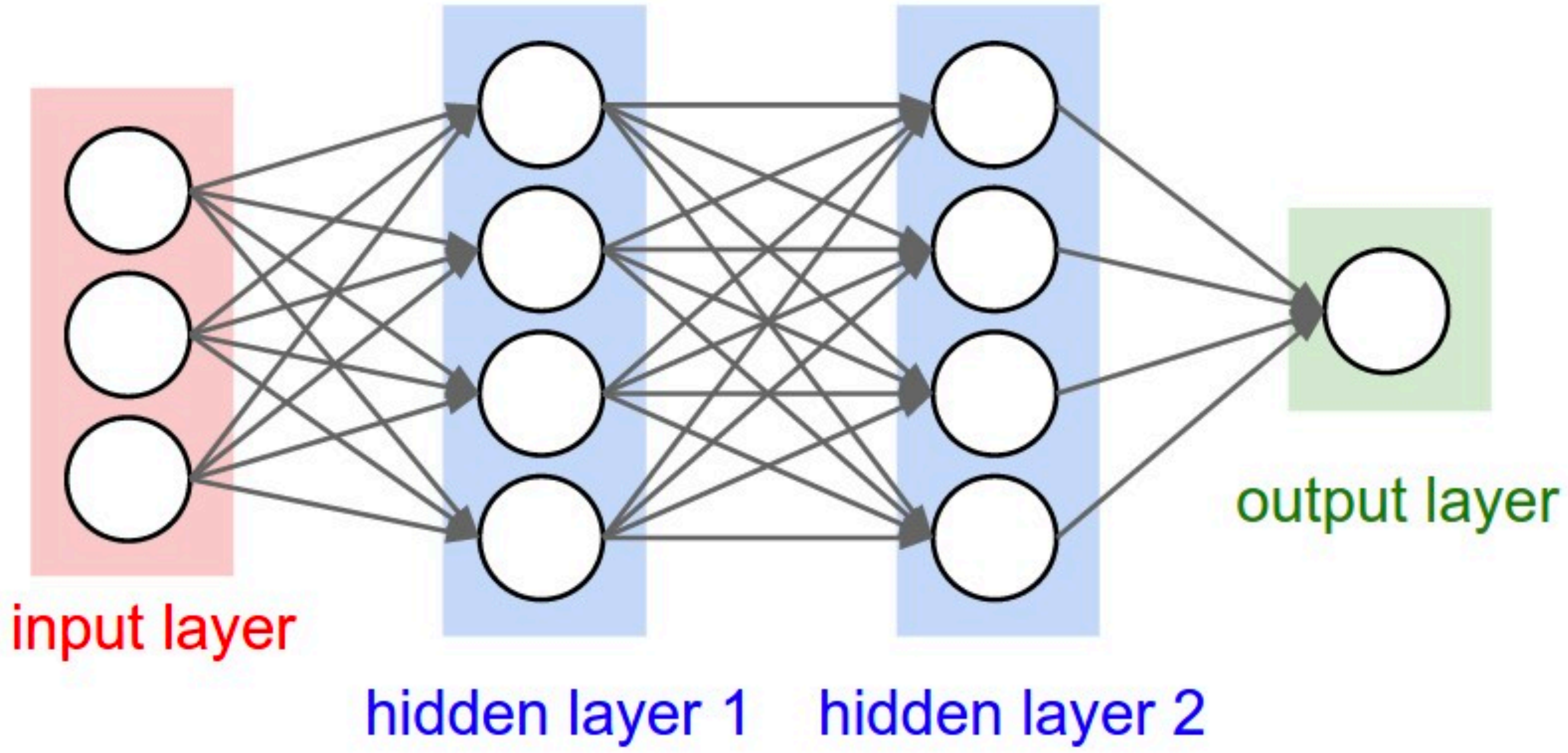


What is Pooling? And Why?



Images from CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2019.

What is Fully Connected Layer (Linear / Dense)?

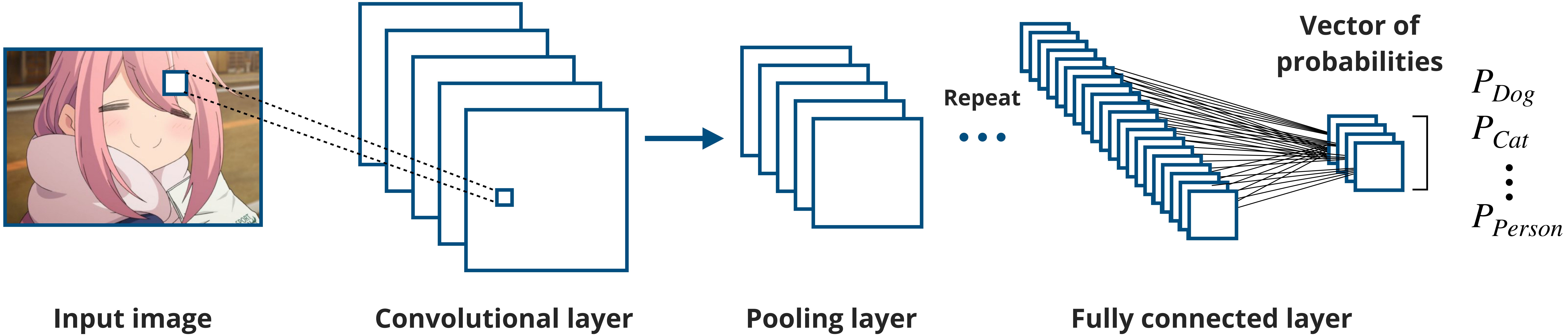


Images from CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2019.

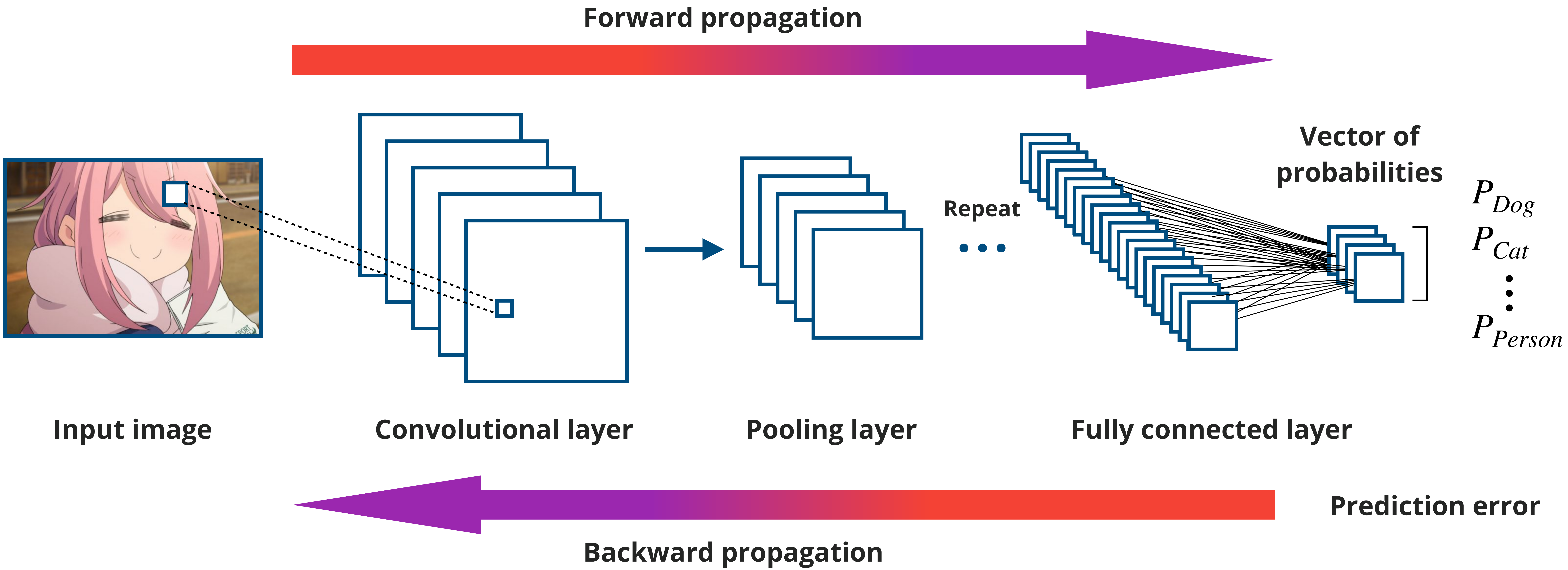
Neurons in a fully connected layer have full connections to all activations in the previous layer.

Their activations can hence be computed with a matrix multiplication followed by a bias offset.

Overview of Convolutional Neural Network (CNN)



Overview of Convolutional Neural Network (CNN)



How to Correct the Prediction? — Loss Function

Classification Loss

Prediction

$P_{Dog}\ P_{Cat}\ \ldots\ P_{Person}$

$[0.14,\ 0.4,\ \ldots,\ 0.05]$

Label (Ground-truth)

$\hat{P}_{Dog}\ \hat{P}_{Cat}\ \cdots\ \hat{P}_{Person}$

$[0,\ 1,\ \ldots,\ 0]$

$$CrossEntropy = -\,(y\log p + (1-y)log(1-p))$$

Regression Loss

Prediction $\hat{y}$

Label (Ground-truth) y

$$MeanSquareError\ (MSE) = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

$$MeanAbsoluteError\ (MAE) = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}$$

How to Correct the Prediction? — Loss Function

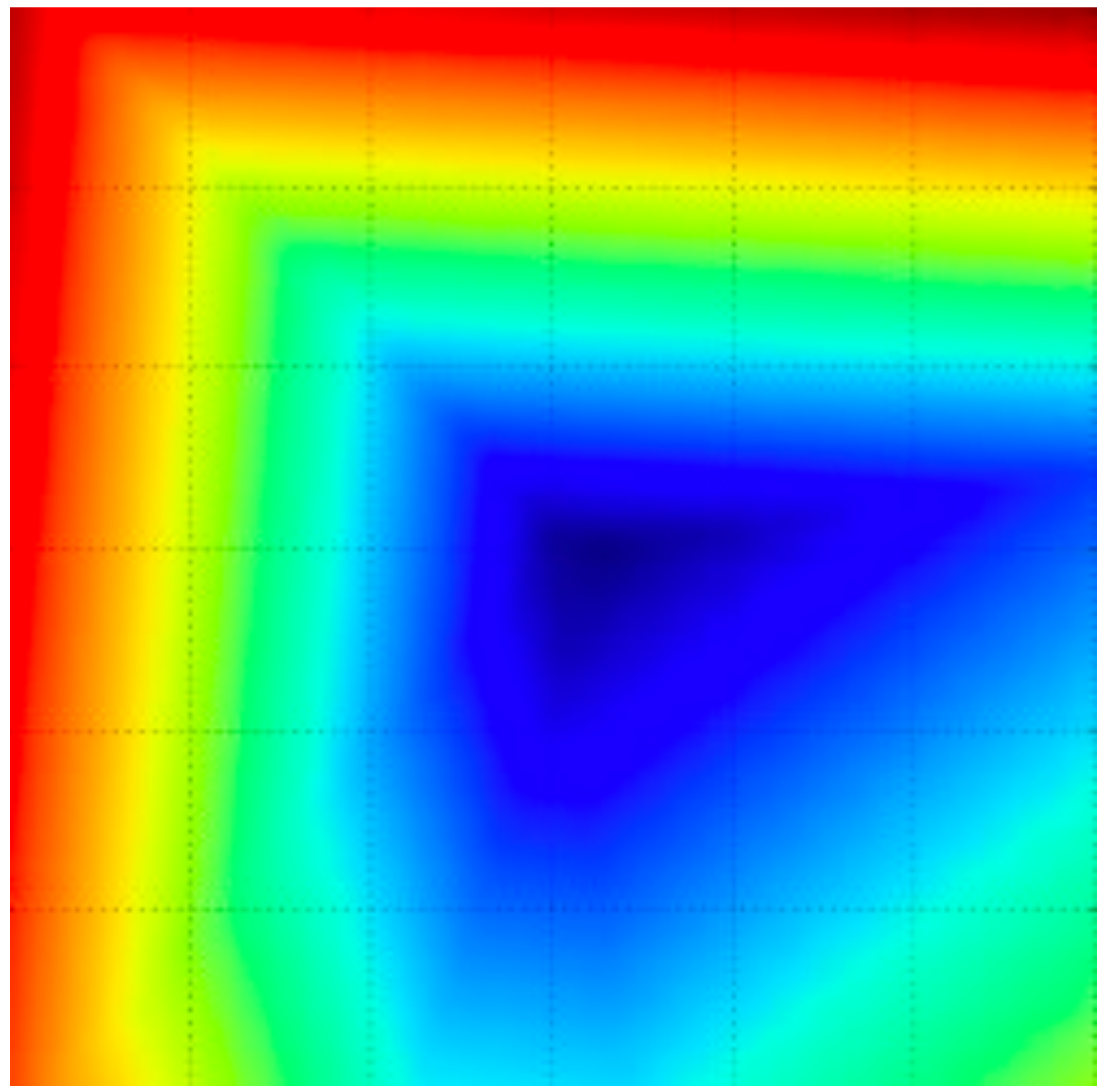
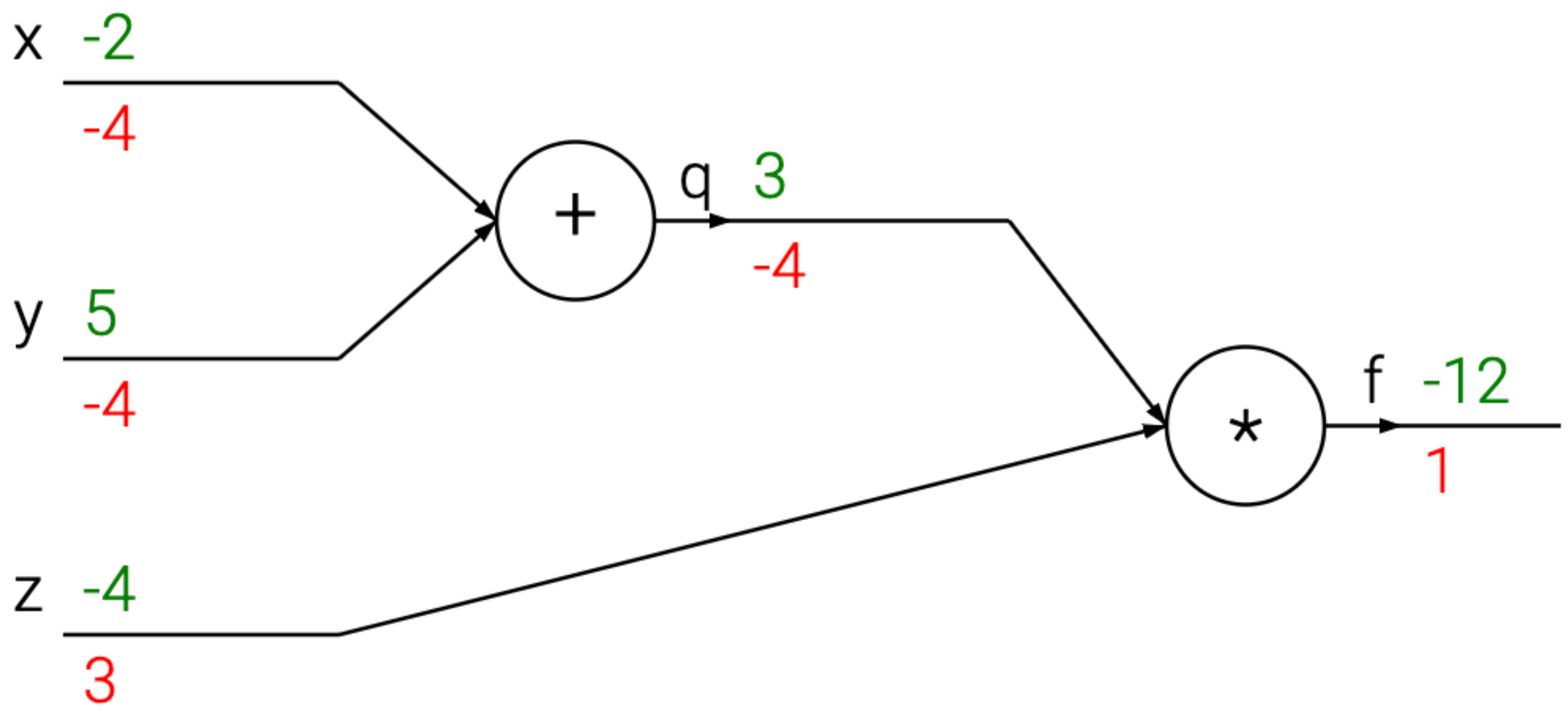


Illustration of loss function landscape,
where **Blue = low loss**, **Red = high loss**

Image from CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2019.

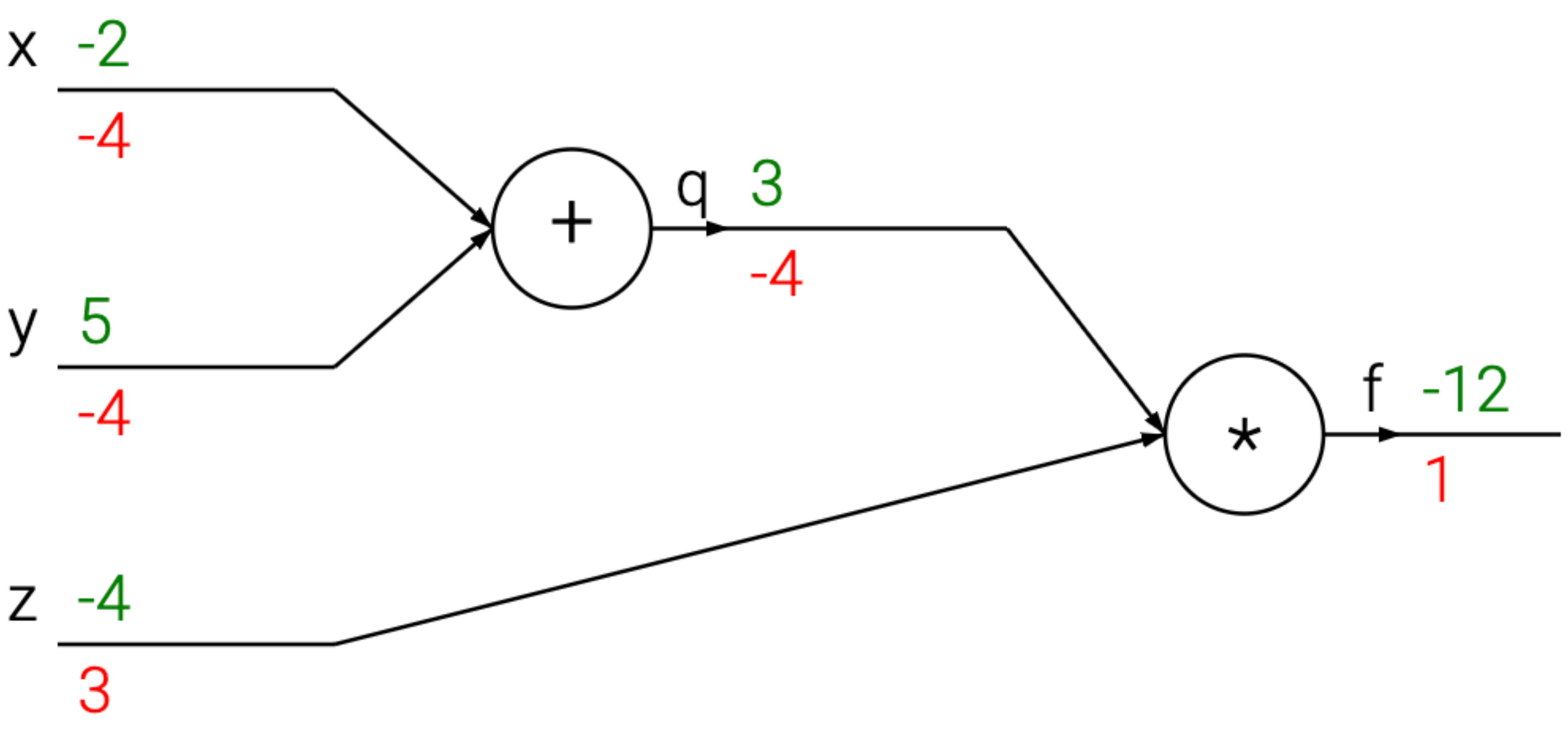
How to Correct the Prediction? — Gradients & Backpropogation



- The **forward pass** computes values from inputs to output (shown in green).
- The **backward pass** then performs **backpropagation** which starts at the end and recursively **applies the chain rule** to **compute the gradients** (shown in red).

Image from CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2019.

How to Correct the Prediction? — Gradients & Backpropogation



$$\frac{df}{dx} = -4$$

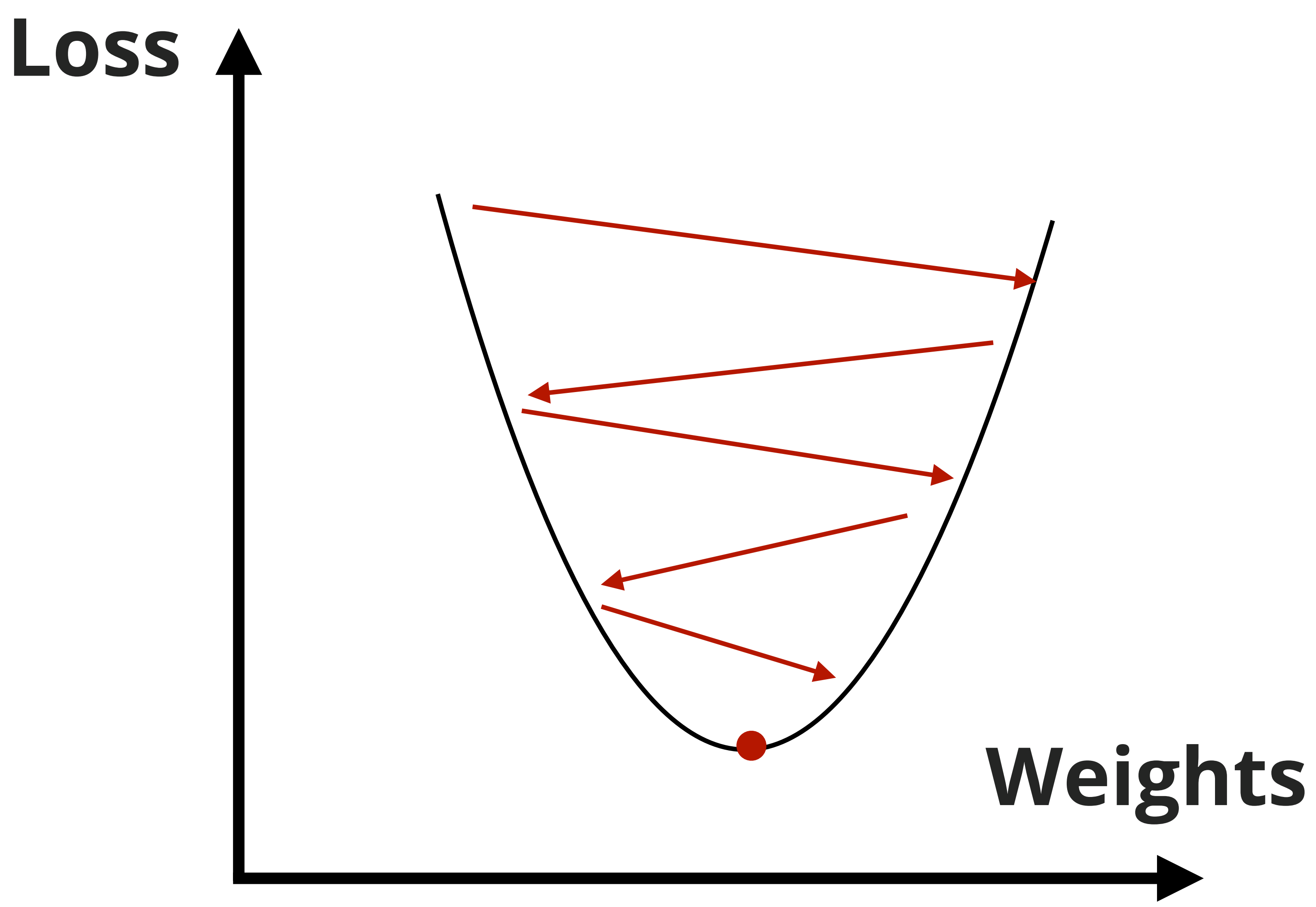
$$\frac{df}{dy} = -4$$

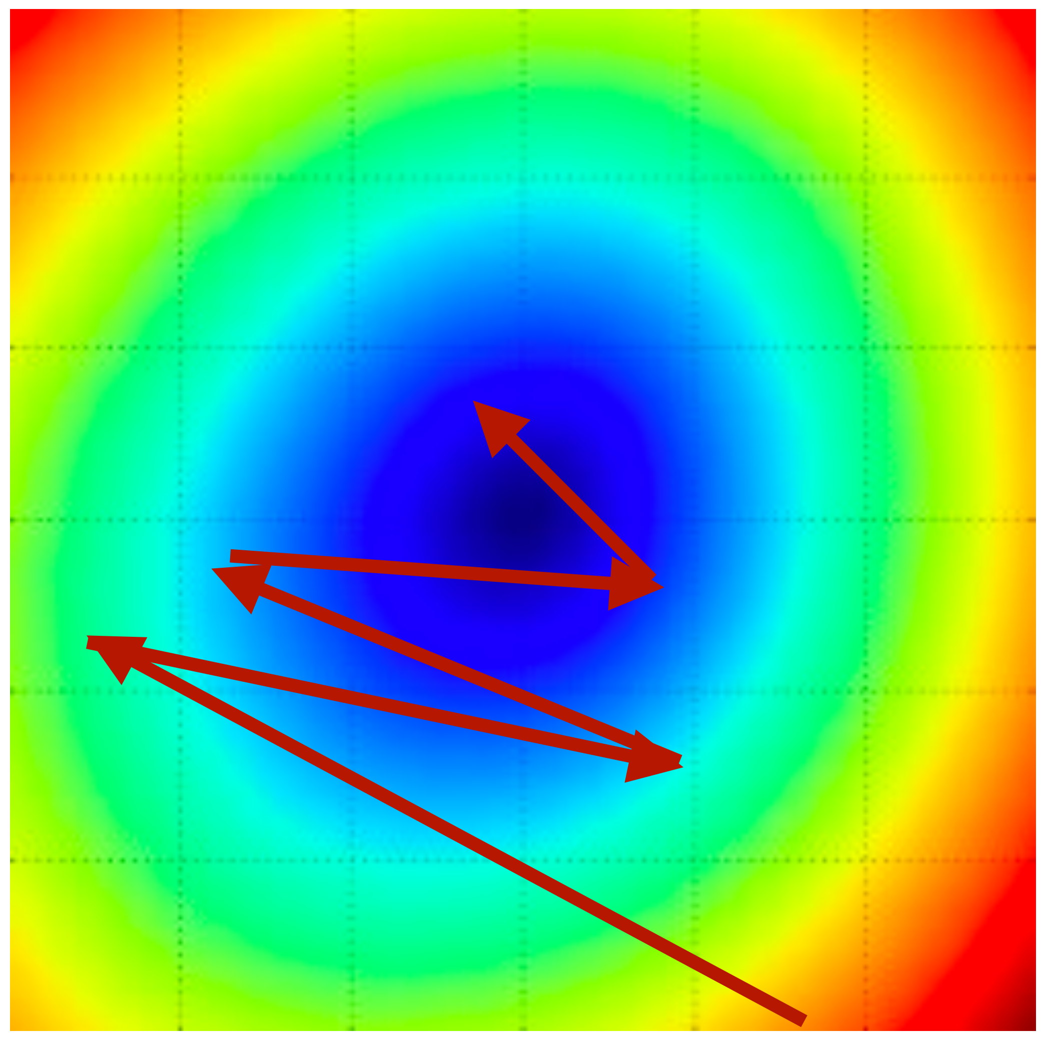
$$\frac{df}{dz} = 3$$

At the end we get **gradients** $\frac{df}{dx}$, $\frac{df}{dy}$, and $\frac{df}{dz}$, which tell us the **sensitivity** of the variables x, y, and z on f.

Image from CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2019.

How to Correct the Prediction? — Gradient Descent



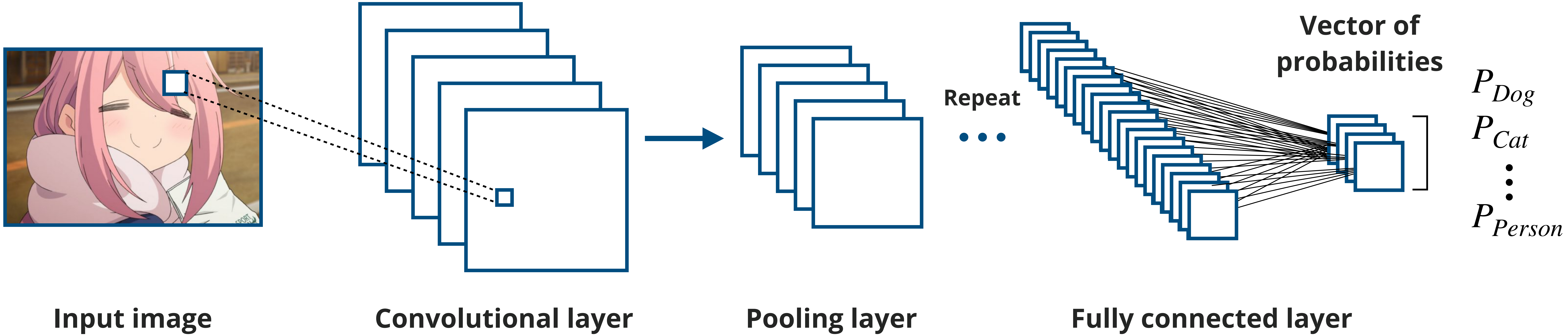


Perform parameter update using the calculated gradients.

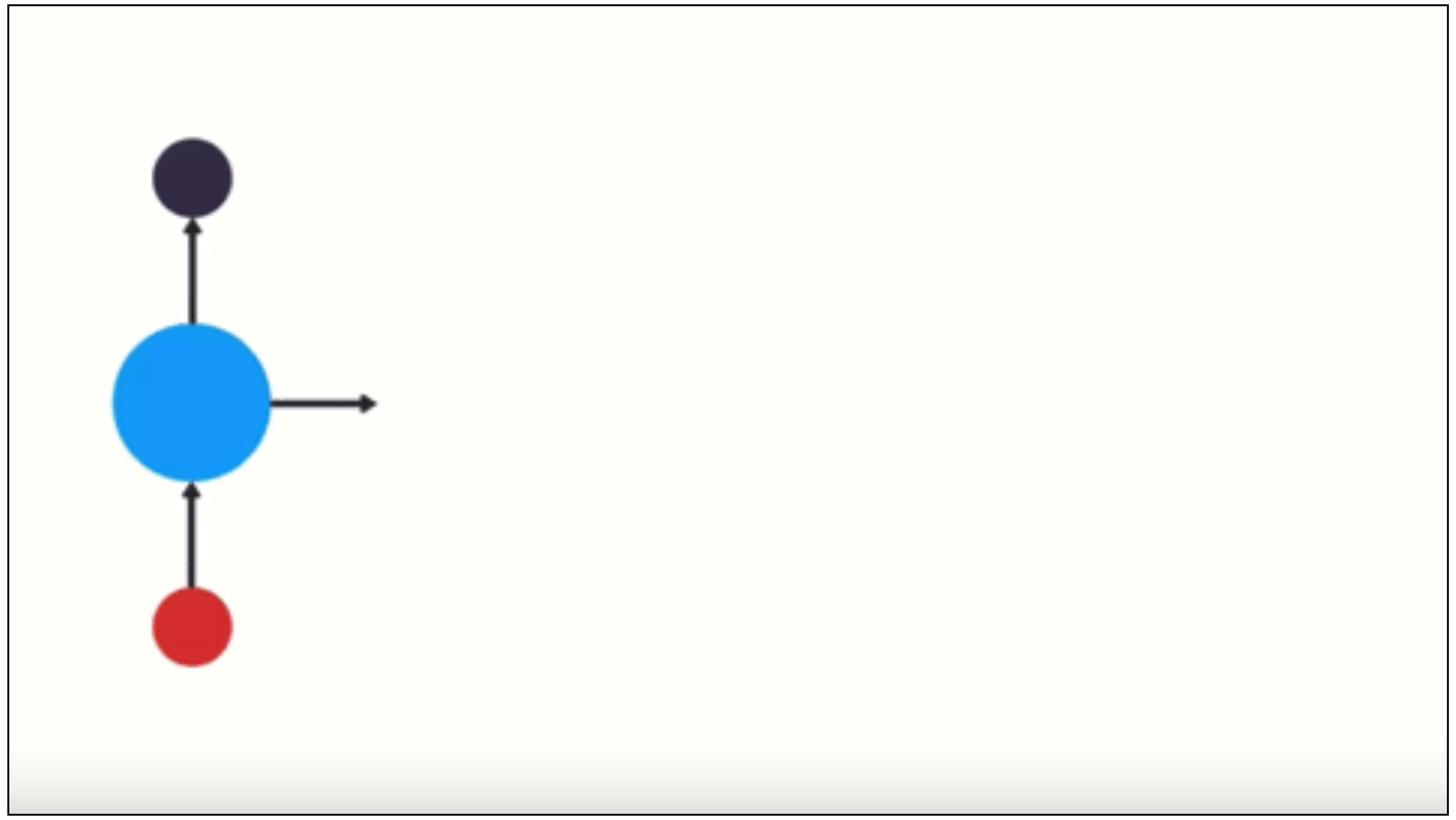
Keywords: mini-batch gradient descent
stochastic gradient descent (SGD)

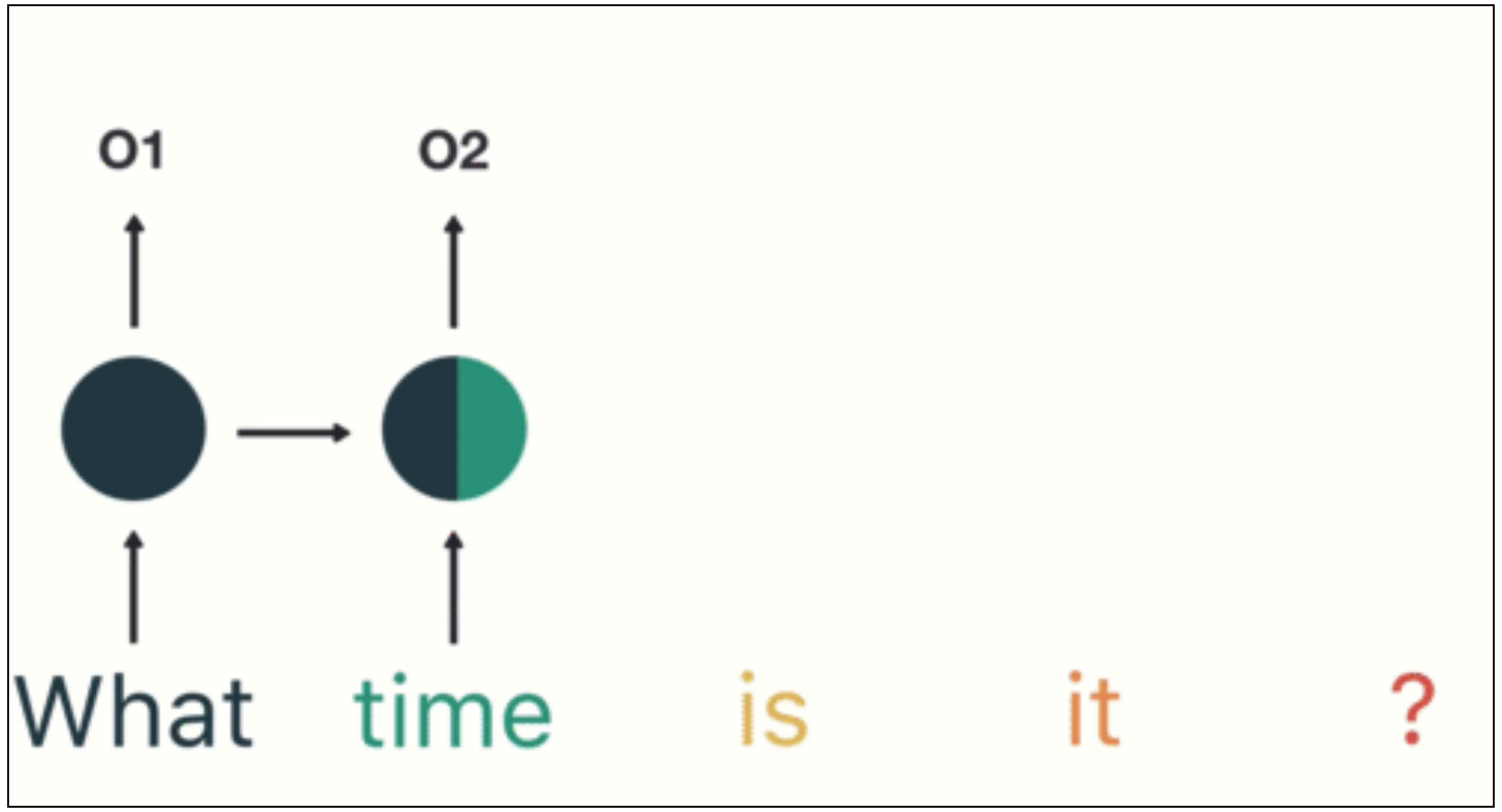
Image from CS231n: Convolutional Neural Networks for Visual Recognition, Spring 2019.

Overview of Convolutional Neural Network (CNN)



Overview of Recurrent Neural Networks (RNNs)





GIF from https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

Outline

- Large State MDP Problem
- Case Study — Self-Driving Car & Introduction to Computer Vision
- Deep Learning Frameworks
- Common Reinforcement Learning Environments

Introduction to Common Deep Learning Frameworks

PyTorch

TensorFlow

Caffe2

mxnet

Deep Learning Frameworks

OpenAI Baselines

Dopamine

rllib

Chainer RL

DeepMind TRFL

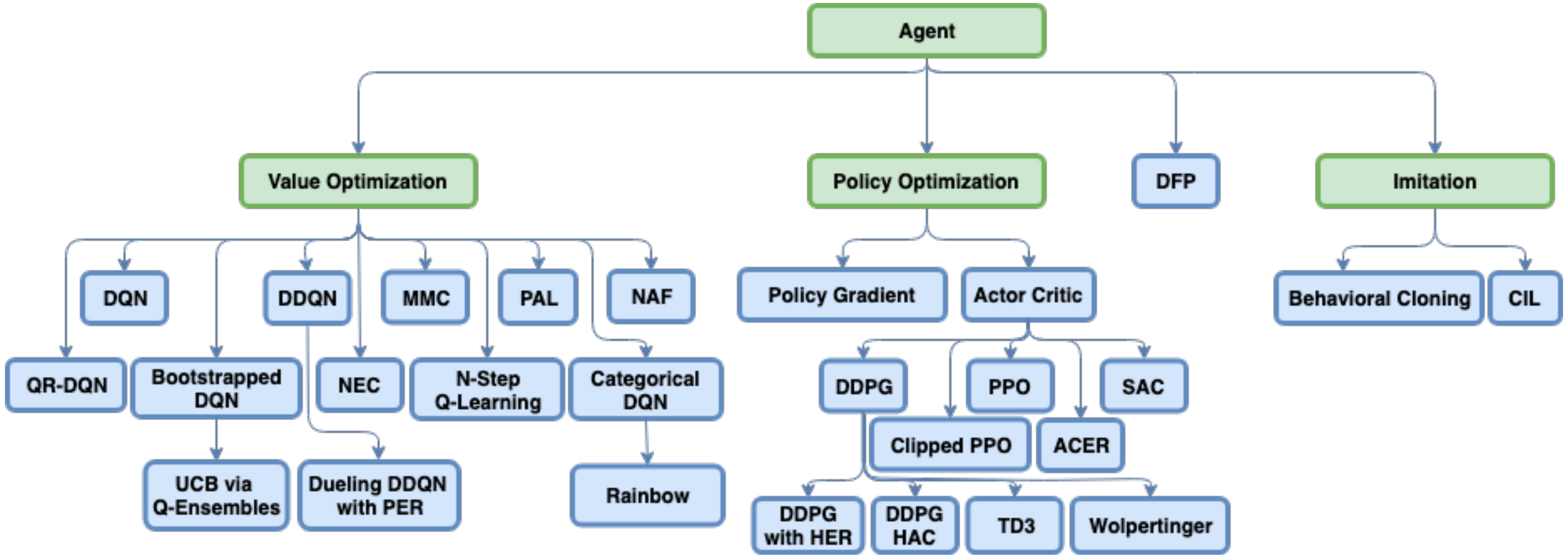
Deep Learning Frameworks

- Based on Tensorflow
- A set of high-quality implementations of reinforcement learning algorithms.

OpenAI Baselines

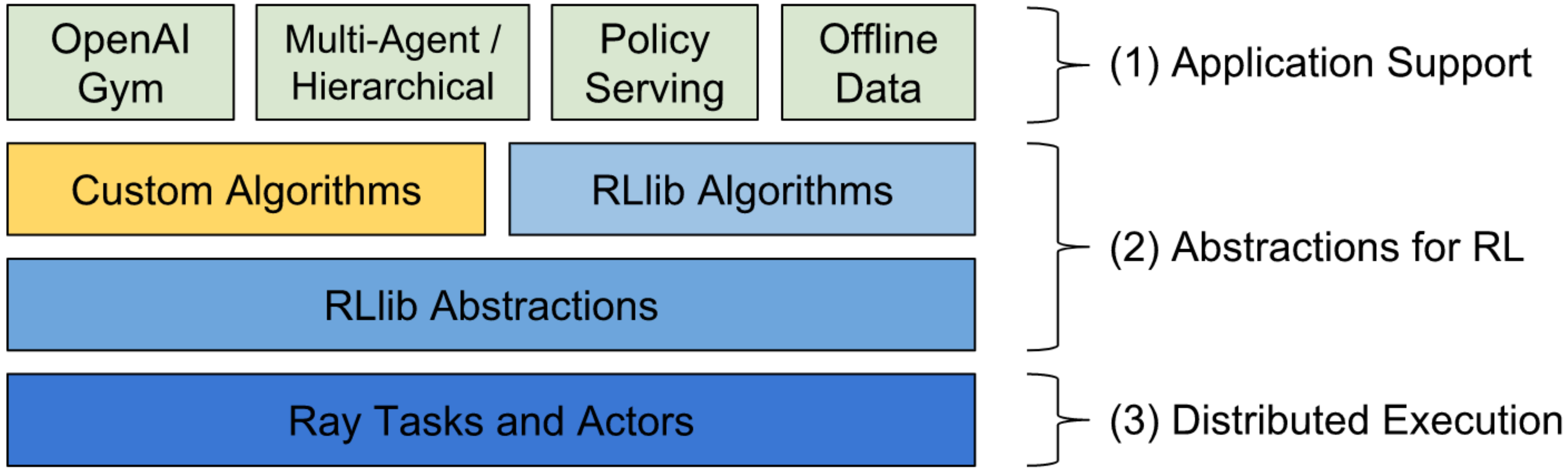
Deep Learning Frameworks

- Based on Tensorflow
- More Algorithm and environment supported



Deep Learning Frameworks

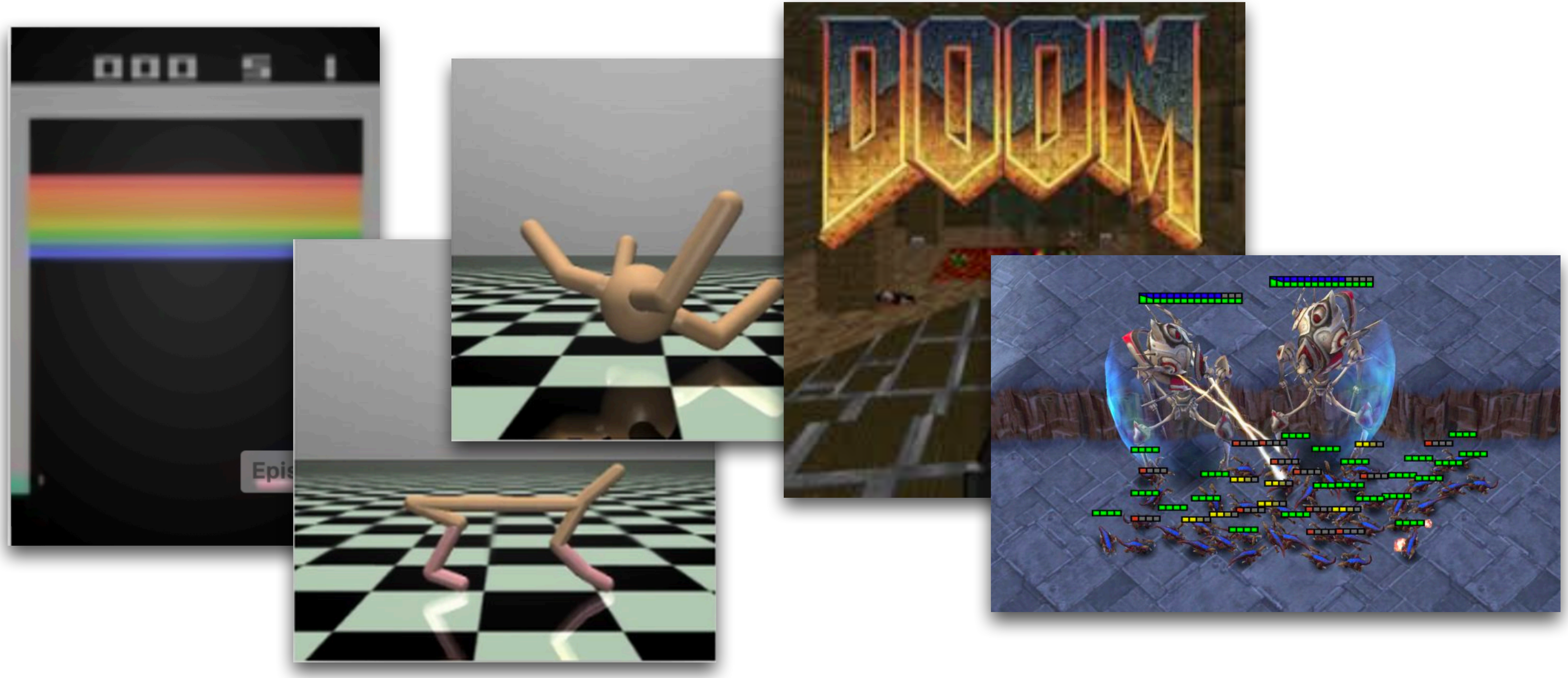
- Support TensorFlow, and PyTorch,
- Support both multi-agent and multi policy training environments
- Support large cluster learning



Outline

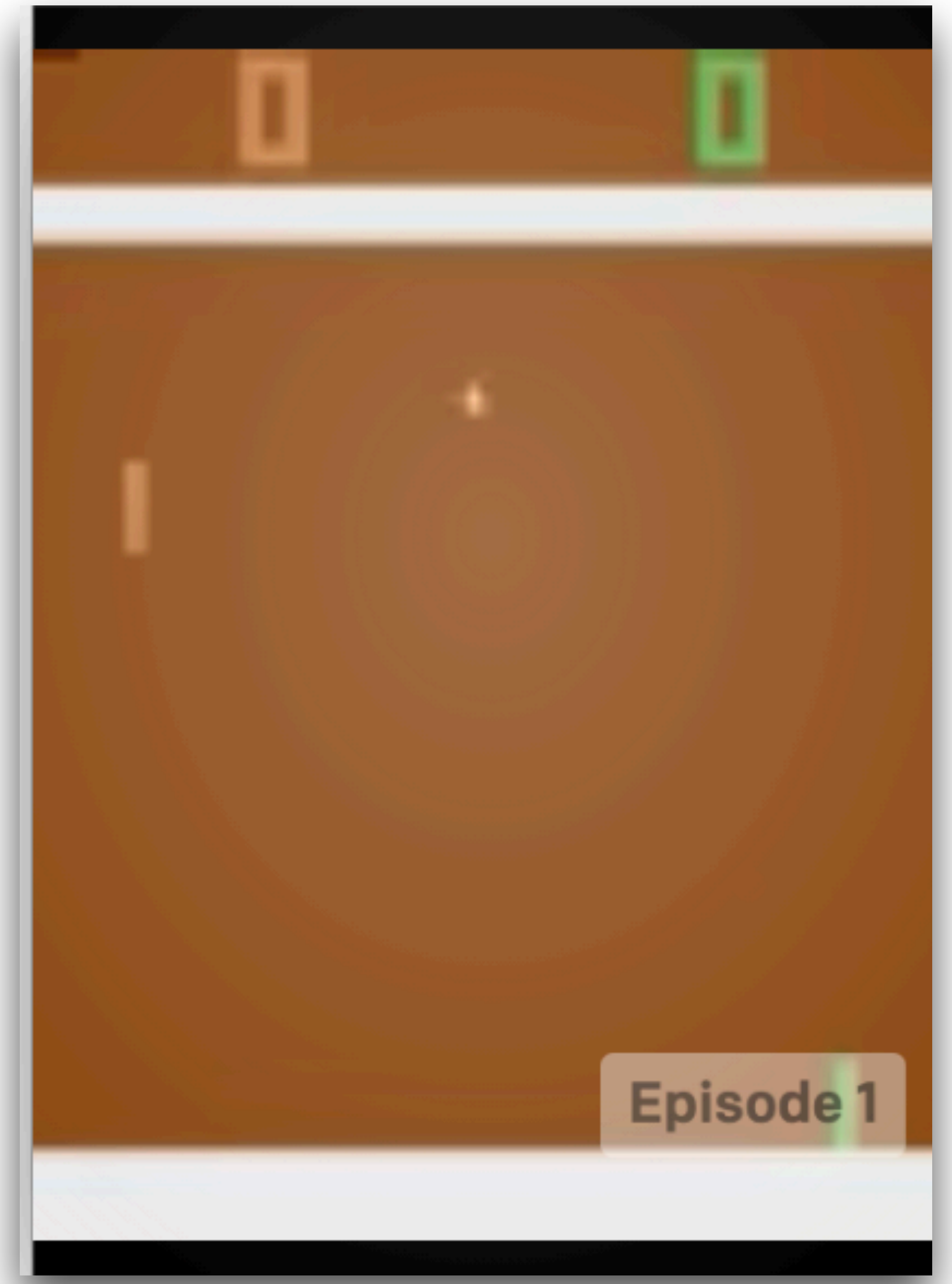
- Large State MDP Problem
- Case Study — Self-Driving Car & Introduction to Computer Vision
- Deep Learning Frameworks
- Common Reinforcement Learning Environments

Common Reinforcement Learning Environment



Atari 2600

- A collection of discrete control game
- Goal: To reach high scores
- **Pong** — An easy game in Atari games
- **Breakout** — Famous game



Pong

Breakout

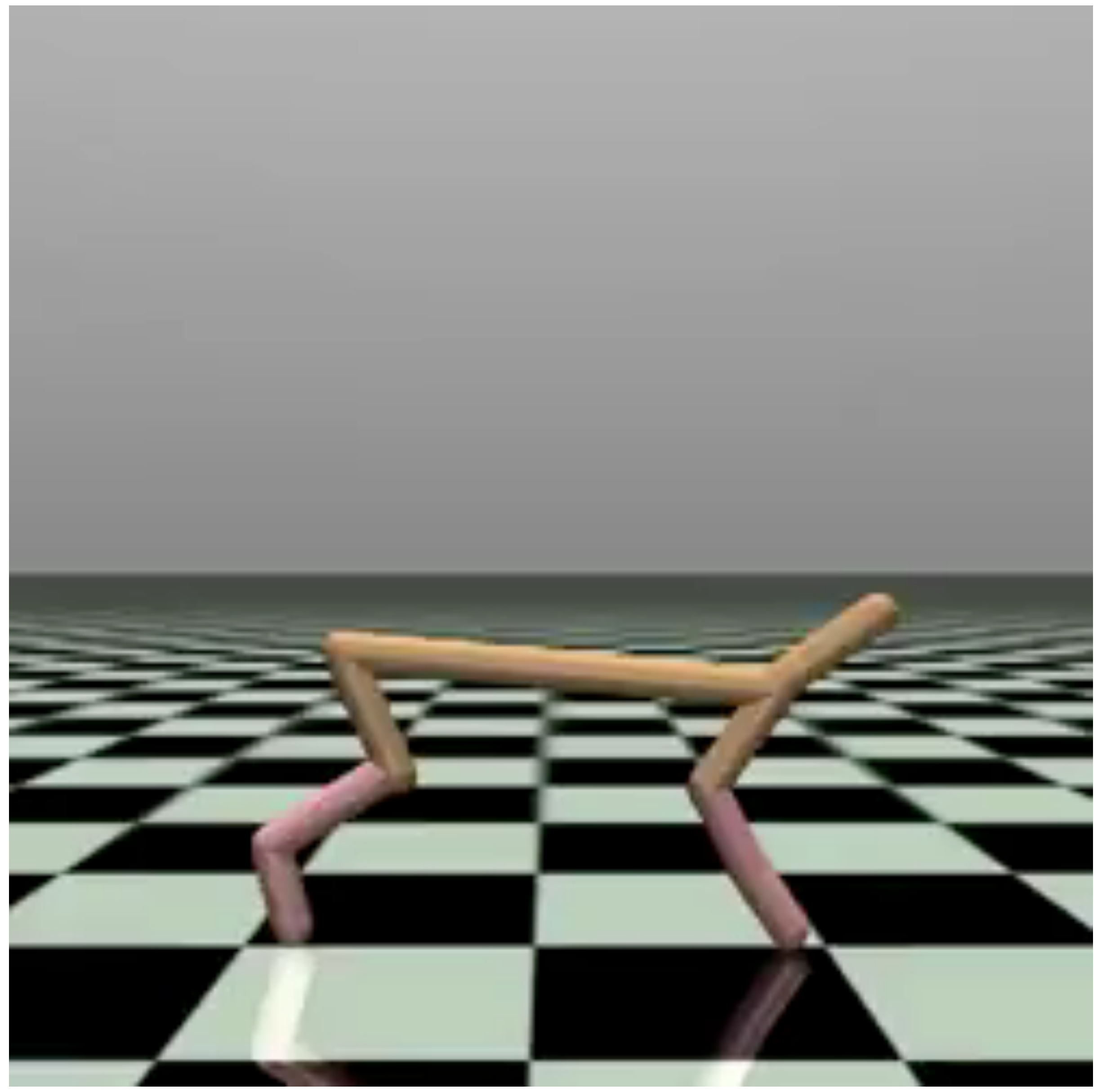
Atari 2600

Easy Exploration			Hard Exploration	
Human-Optimal		Score Exploit	Dense Reward	Sparse Reward
Assault	Asterix	Beam Rider	Alien	Freeway
Asteroids	Atlantis	Kangaroo	Amidar	Gravitar
Battle Zone	Berzerk	Krull	Bank Heist	Montezuma's Revenge
Bowling	Boxing	Kung-fu Master	Frostbite	Pitfall!
Breakout	Centipede	Road Runner	H.E.R.O.	Private Eye
Chopper Cmd	Crazy Climber	Seaquest	Ms. Pac-Man	Solaris
Defender	Demon Attack	Up n Down	Q*Bert	Venture
Double Dunk	Enduro	Tutankham	Surround	
Fishing Derby	Gopher		Wizard of Wor	
Ice Hockey	James Bond		Zaxxon	
Name this Game	Phoenix			
Pong	River Raid			
Robotank	Skiing			
Space Invaders	Stargunner			

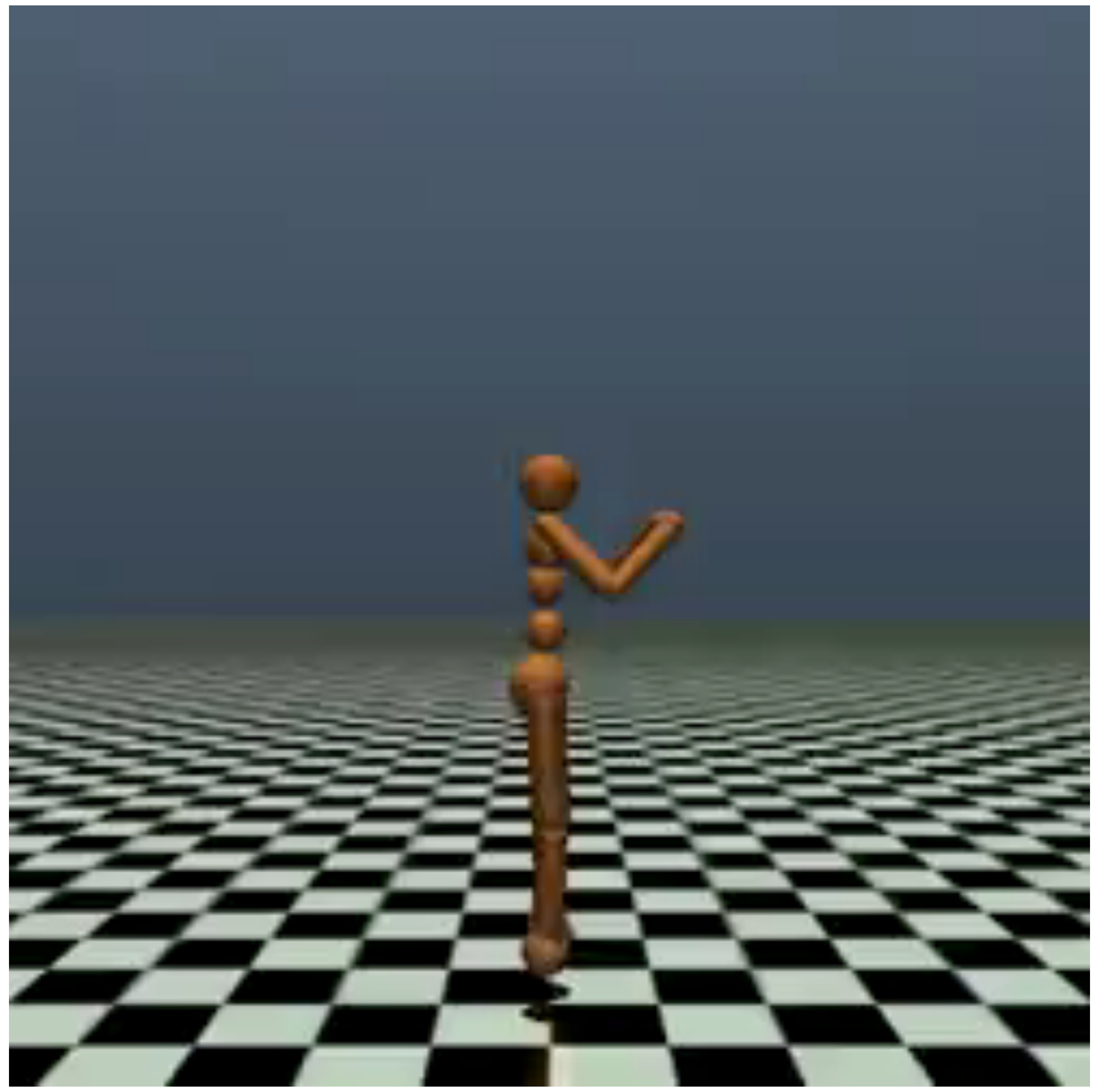
A rough taxonomy of Atari 2600 games according to their exploration difficulty. (M. G. Bellemare, 2016)

Mujoco

- A physics engine aiming to facilitate research and development in robotics, biomechanics, graphics and animation.
- Demo



HalfCheetah



Humanoid

DOOM

- Doom-based AI Research Platform for Reinforcement Learning from Raw Visual Information.
- 3D environments, multiple tasks
- Demo

StarCraft

- A StarCraft environment for collaborative multi-agent reinforcement learning (MARL).
- Demo

Carla

- An open-source simulator for autonomous driving research
- Support development, training, and validation of autonomous driving systems
- Demo

ありがとう
ございます♥