Introduction to Machine Learning programming on Apple devices using CoreML kit.

Apple Foundation Program

Università degli Studi di Napoli Parthenope

Foundation Program

Michele Di Capua, PhD

- Course IoT & Lab (DIST Parthenope)
- Teacher at Apple Foundation Program (Parthenope)
- Contacts: micheledicapua@gmail.com

Agenda

- Apple Foundation Program
- AI & ML intro
- How machine learns?
- Computer vision
- Artificial Neural Networks principles
- AI on smart devices

Apple Foundation Program

Who we are?

Università degli Studi di Napoli Parthenope

Foundation Program

Meeting Foundation Campania - 29 Settembre 2022

a wonderful site

a new learning space

CBL

Collaboration

Prototyping

Coding

Business

UX / UI design

topics

in 7 years…

> 1.500 students

> 60 courses

- iOS
- watchOS
- tvOs

Courses Insights

4 weeks / 80 hrs

BASIC

CBL

Basic Swift
UI/UX
Advanced kits
- ARKit
- SpriteKit
- CoreML

\- …

Final app (iOS)

4 weeks / 80 hrs

ADVANCED

CBL

Advanced Swift
UI/UX
…
watchOS
tvOS
Machine Learning

No final app

4 weeks / 80 hrs

CHALLENGE

CBL

Frameworks
UI/UX (refined)
…
Final app
(iOS, watchOS, tvOS)

Università degli Studi di Napoli Parthenope

Foundation Program

Artificial Intelligence

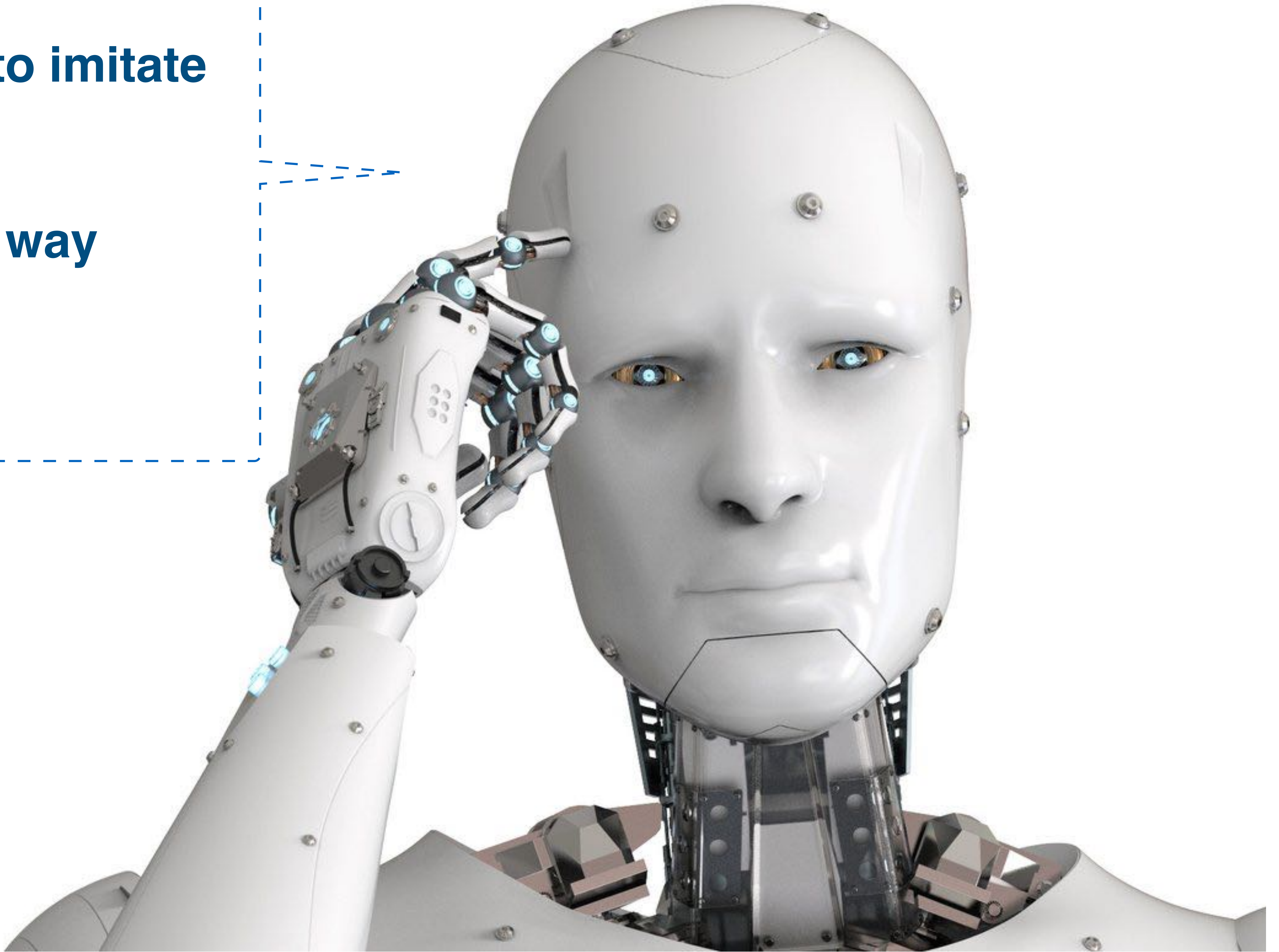
What is a general definition of artificial intelligence ?

AI possible definitions:

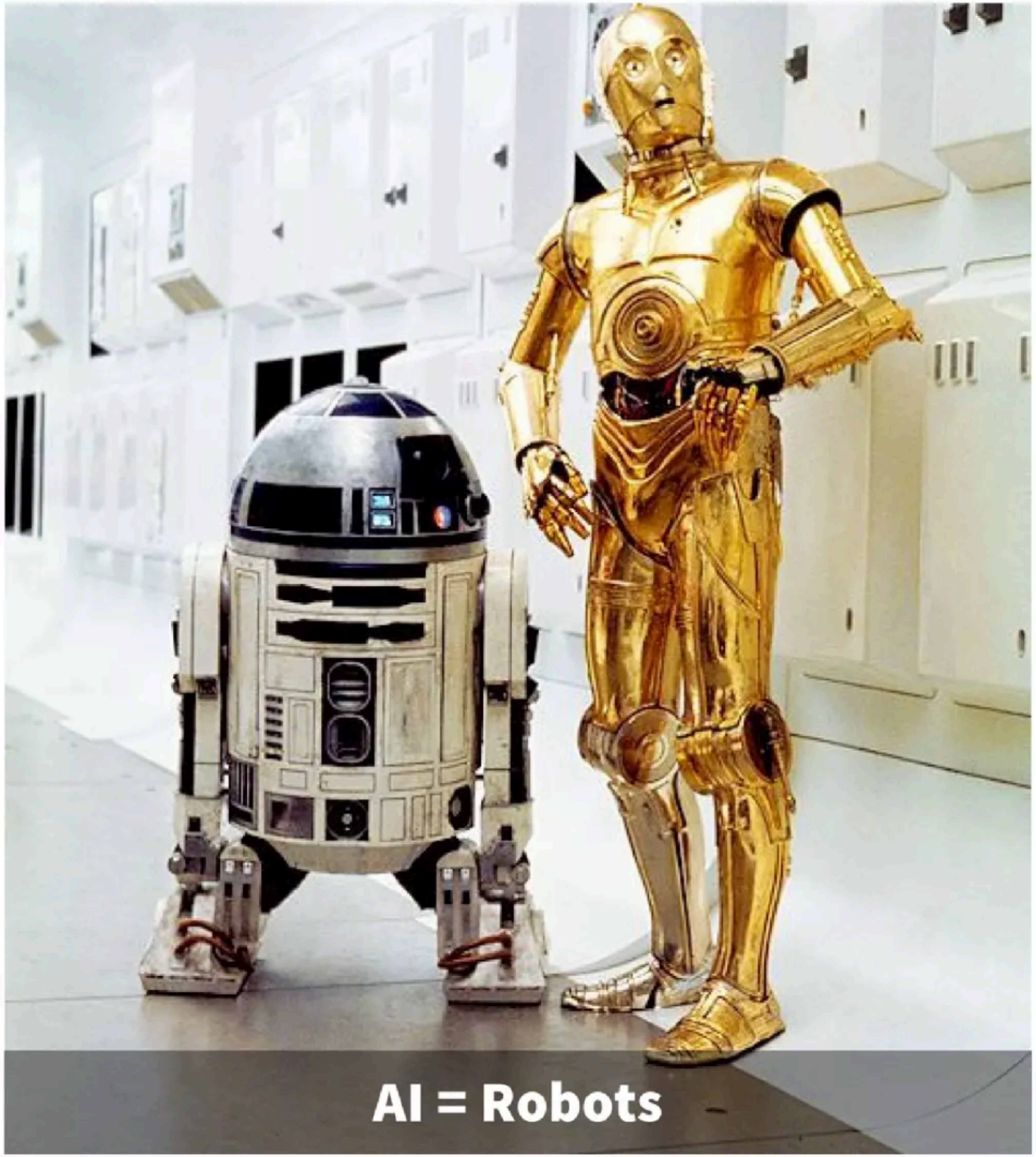
- The capability of a machine to imitate intelligent human behavior

- Solving problems in a smart way

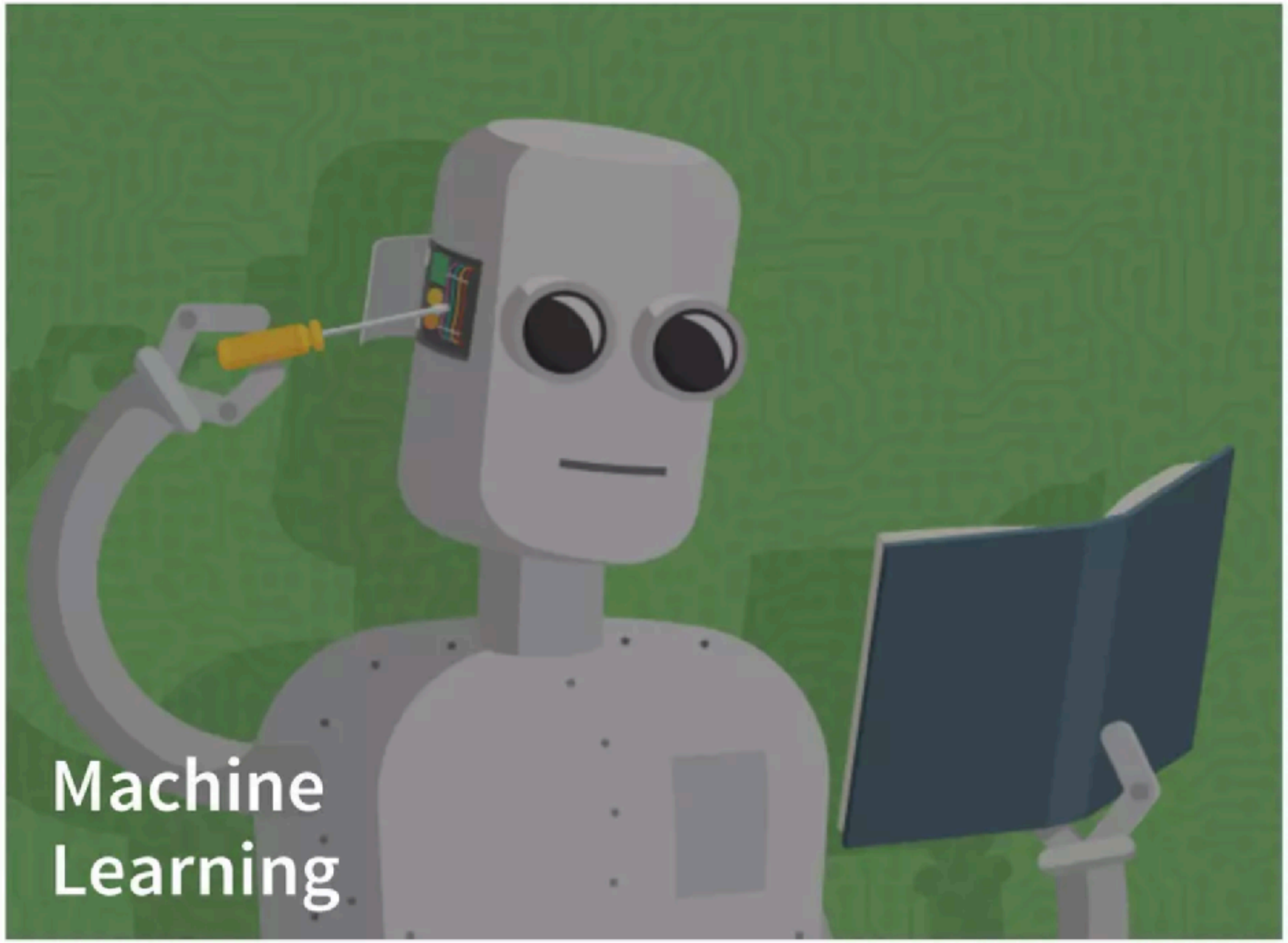
- ...

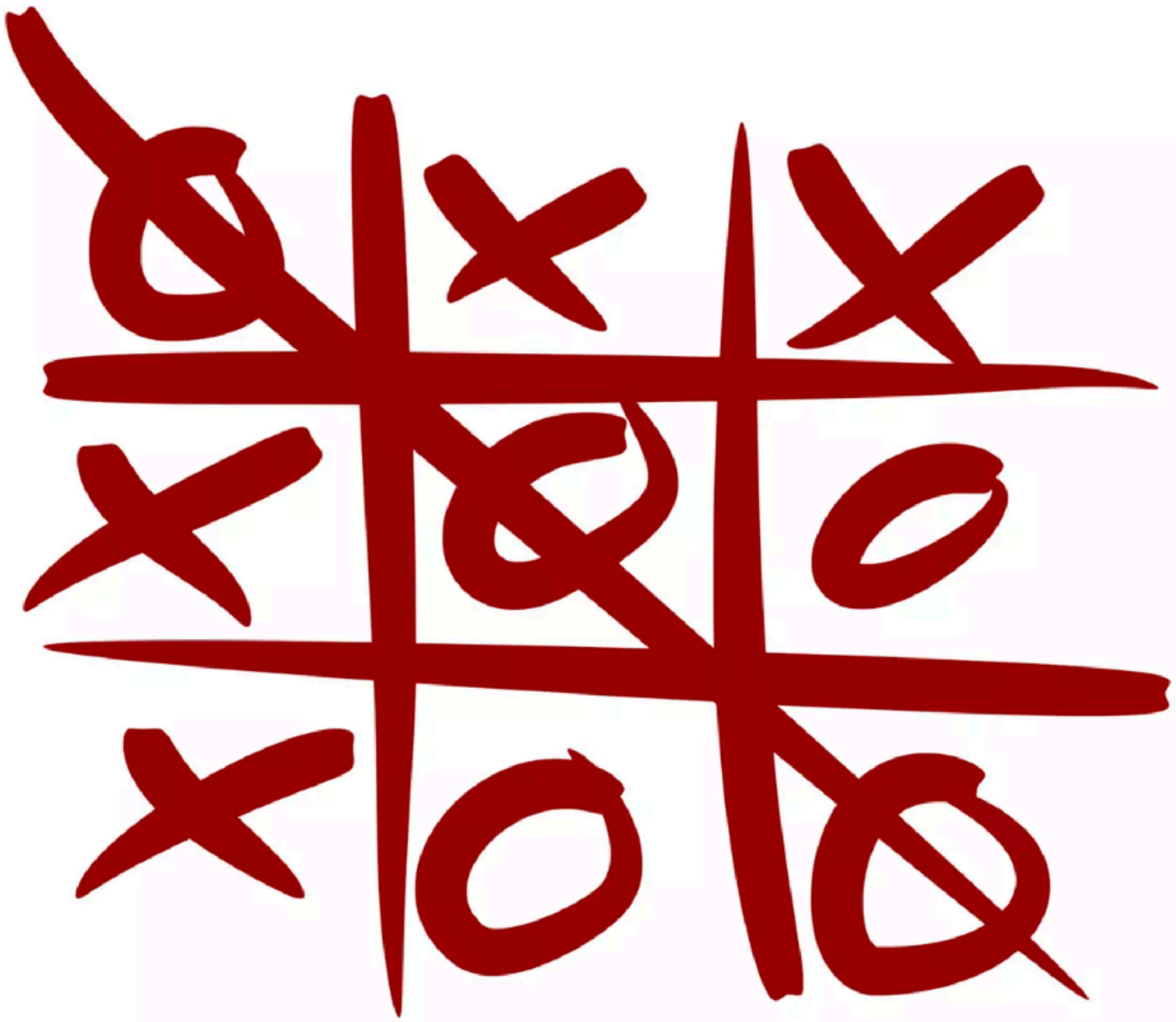



General Perspective

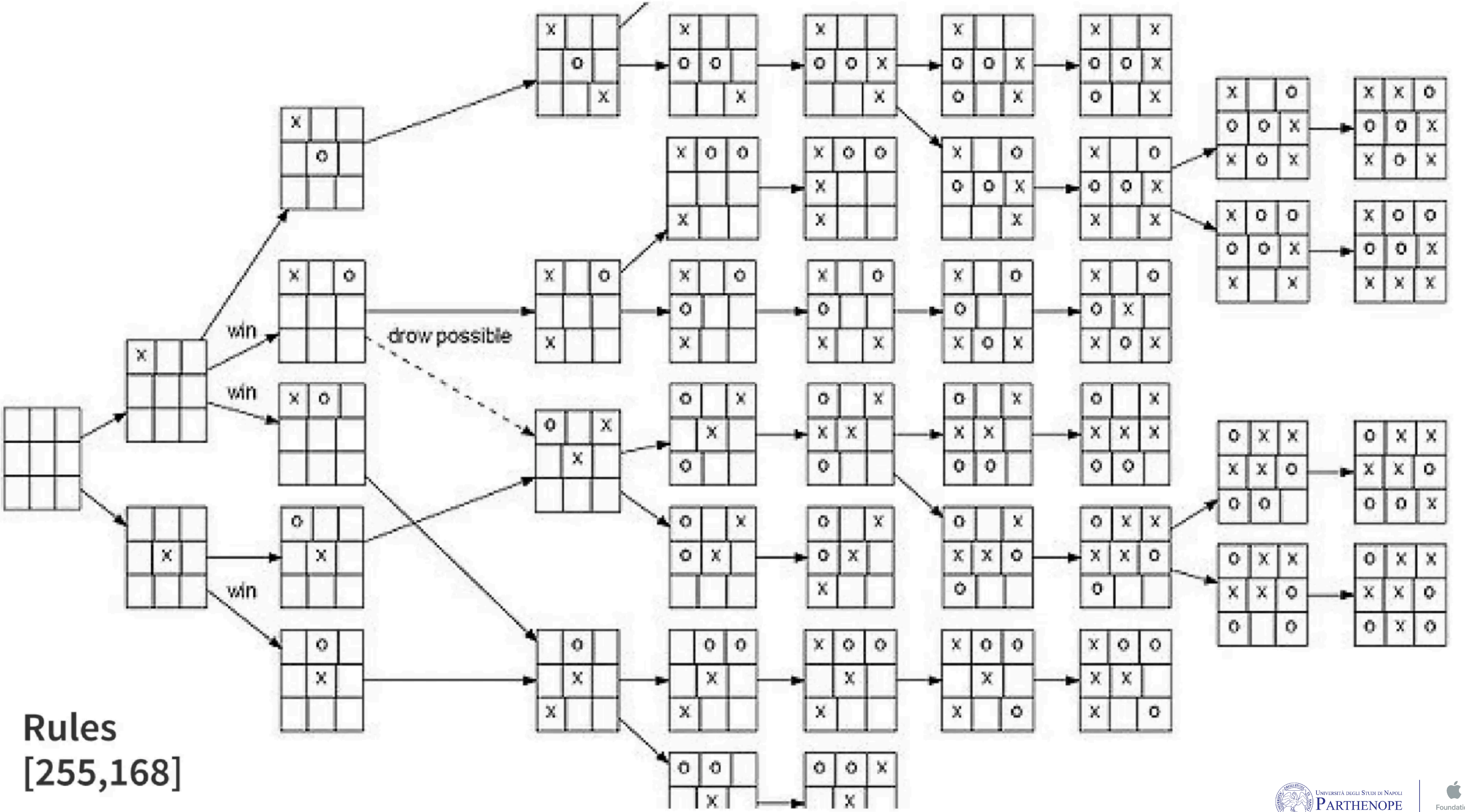

AI = Robots

Solving problems approaches





WAR GAMES (movie, 1983)



Rules
[255,168]

S

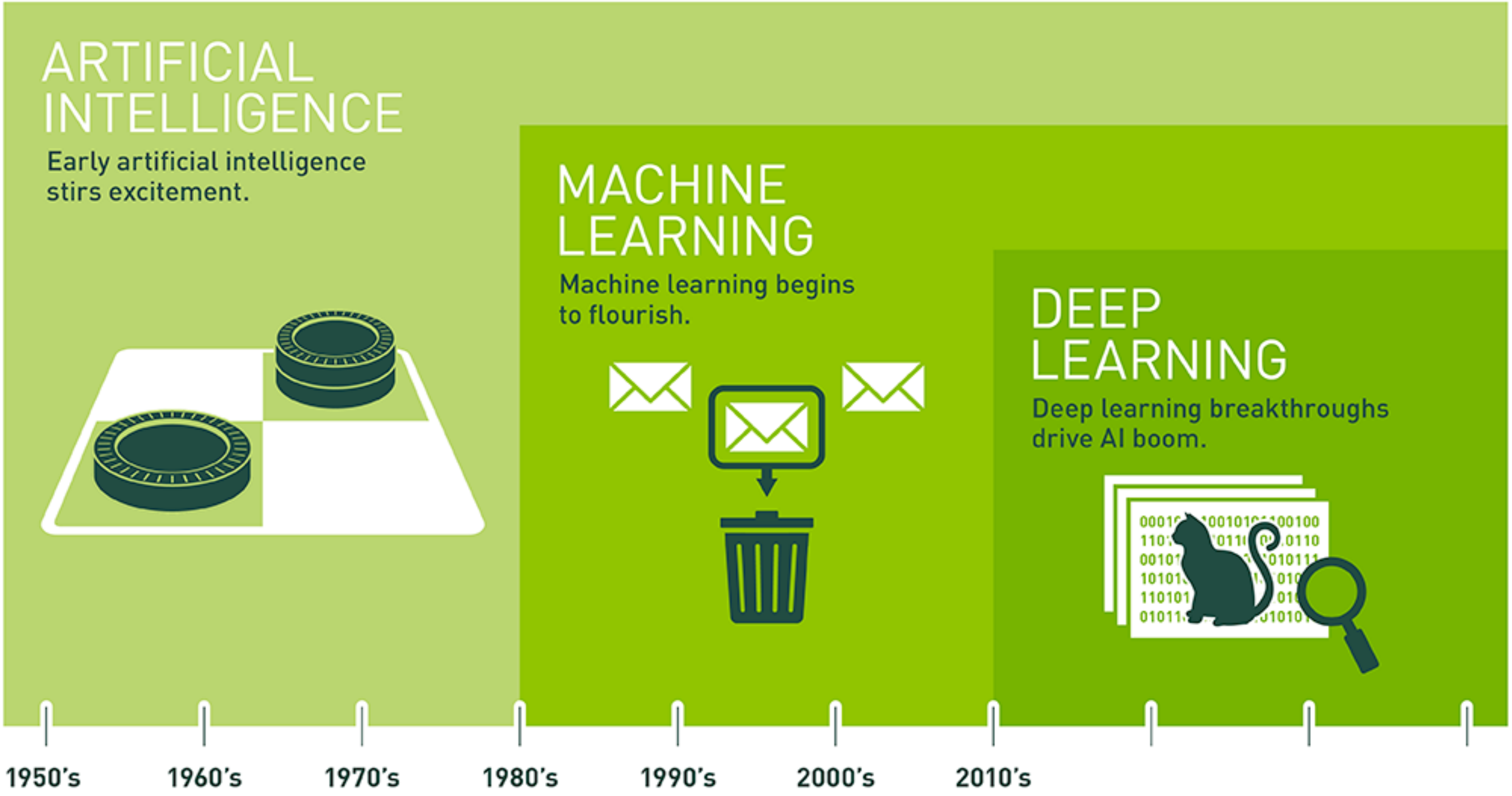
GREETINGS PROFESSOR FALKEN

HELLO

A STRANGE GAME.
THE ONLY WINNING MOVE IS
NOT TO PLAY.

Artificial Intelligence (AI) is the big set. Machine learning (ML) is a subset of AI. Deep learning and shallow learning is a subset of ML.

AI

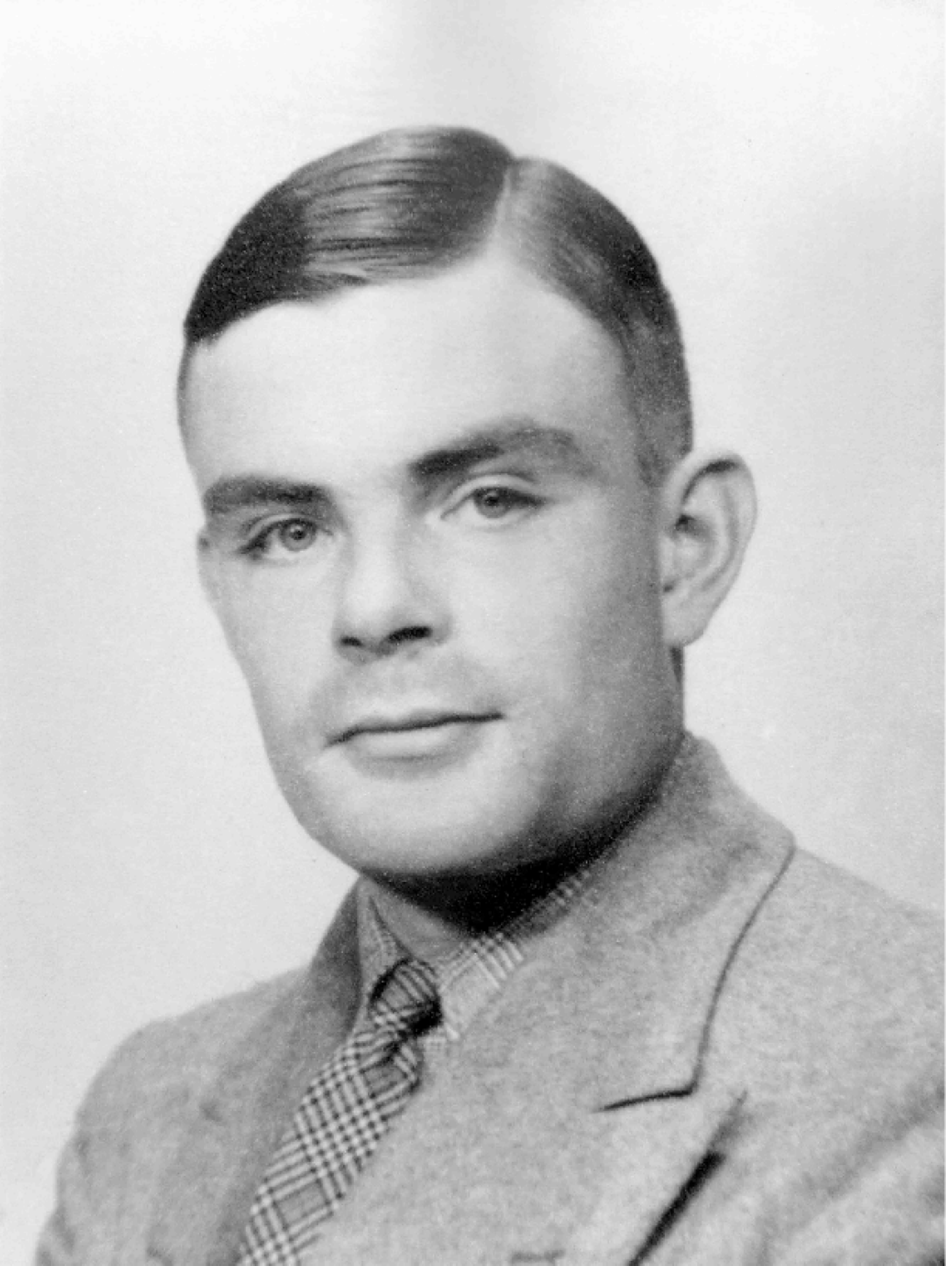


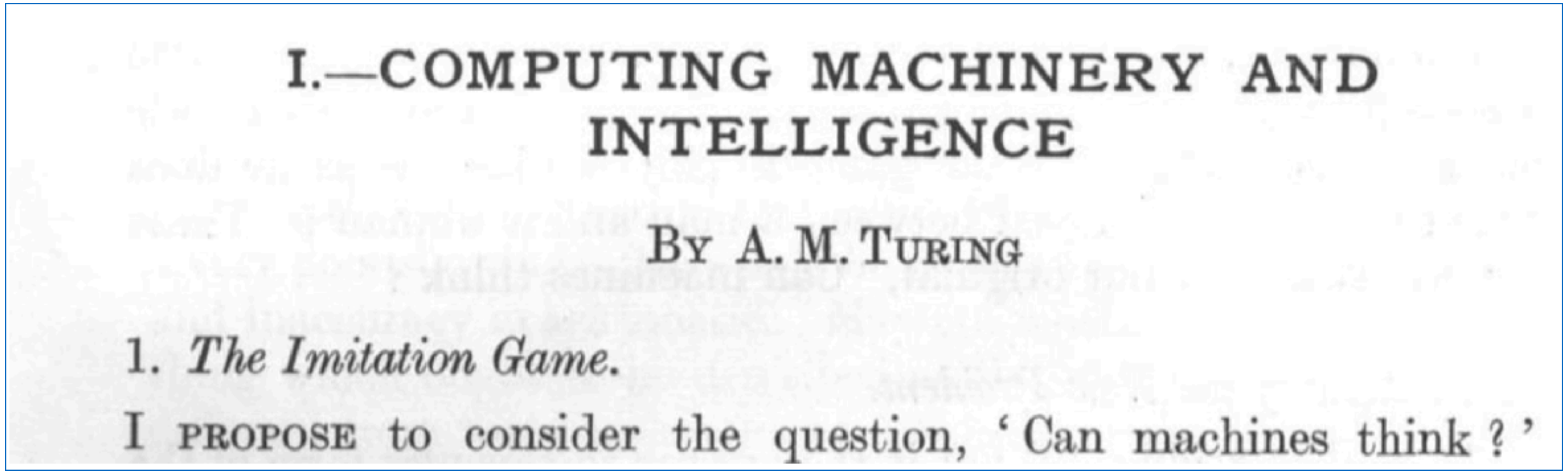
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Università degli Studi di Napoli Parthenope

Foundation Program

Machine Learning





I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

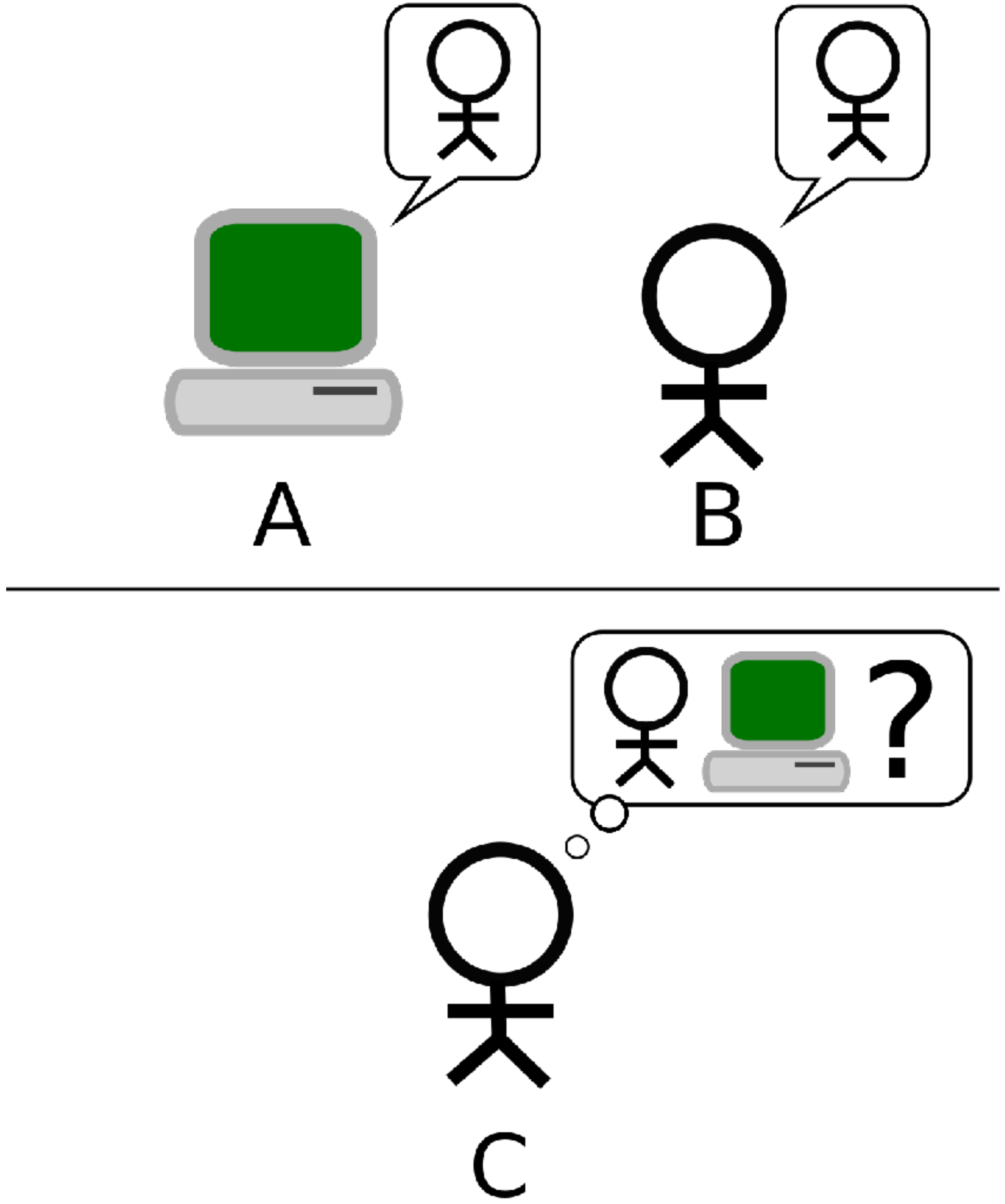
I PROPOSE to consider the question, 'Can machines think?'

Alan Turing, 1950

In the process of trying ***to imitate an adult human mind*** *we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice 3 components,*

a. The initial state of the mind, say at birth,
b. The education to which it has been subjected,
c. Other ***experience****, not to be described as education, to which it has been subjected.*

The Turing Test

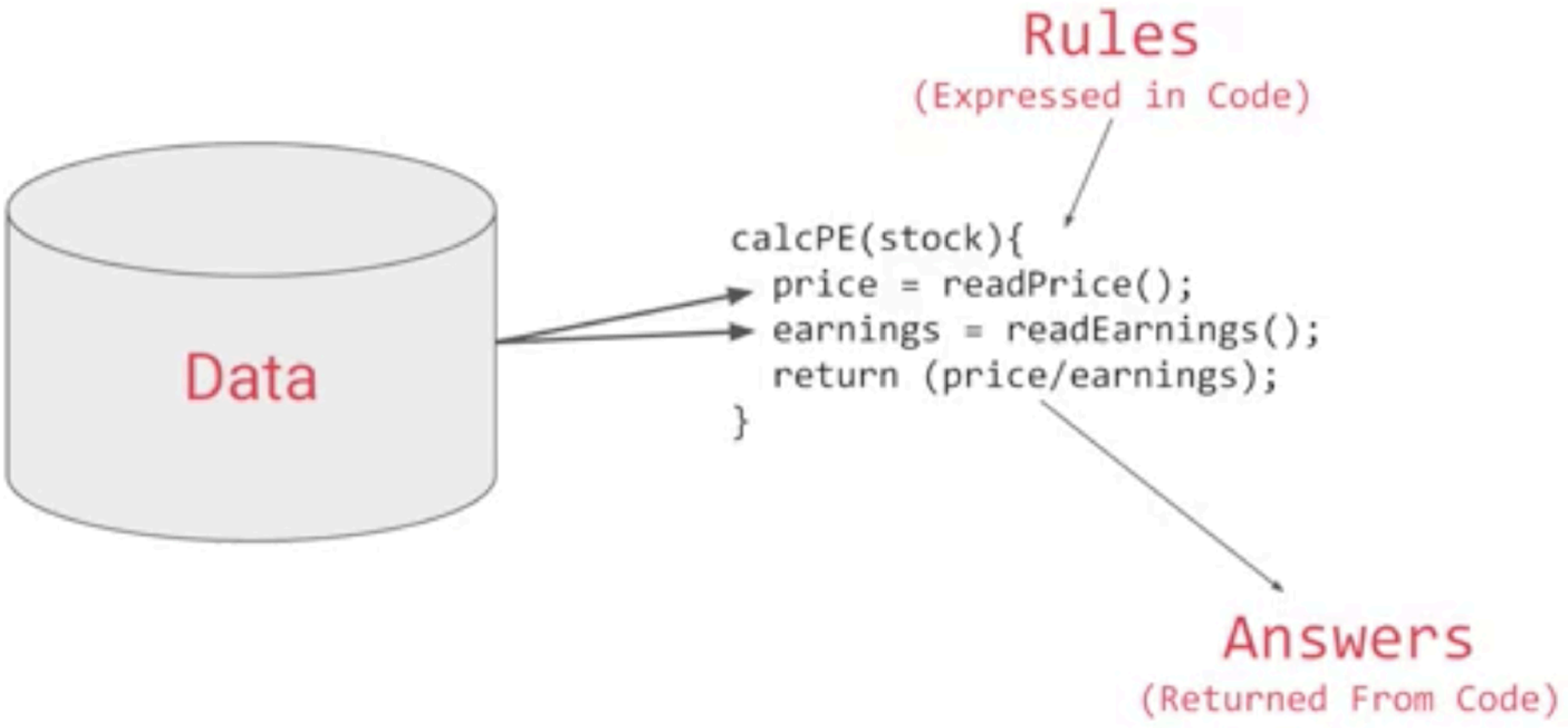


What is the definition of learning ?

*"We define **learning as the transformative process of taking in information** that—when internalized and mixed with what we have **experienced**—changes what we know and builds on what we do. It's based on input, process, and reflection. It is what changes us."*

From The New Social Learning by Tony Bingham and Marcia Conner

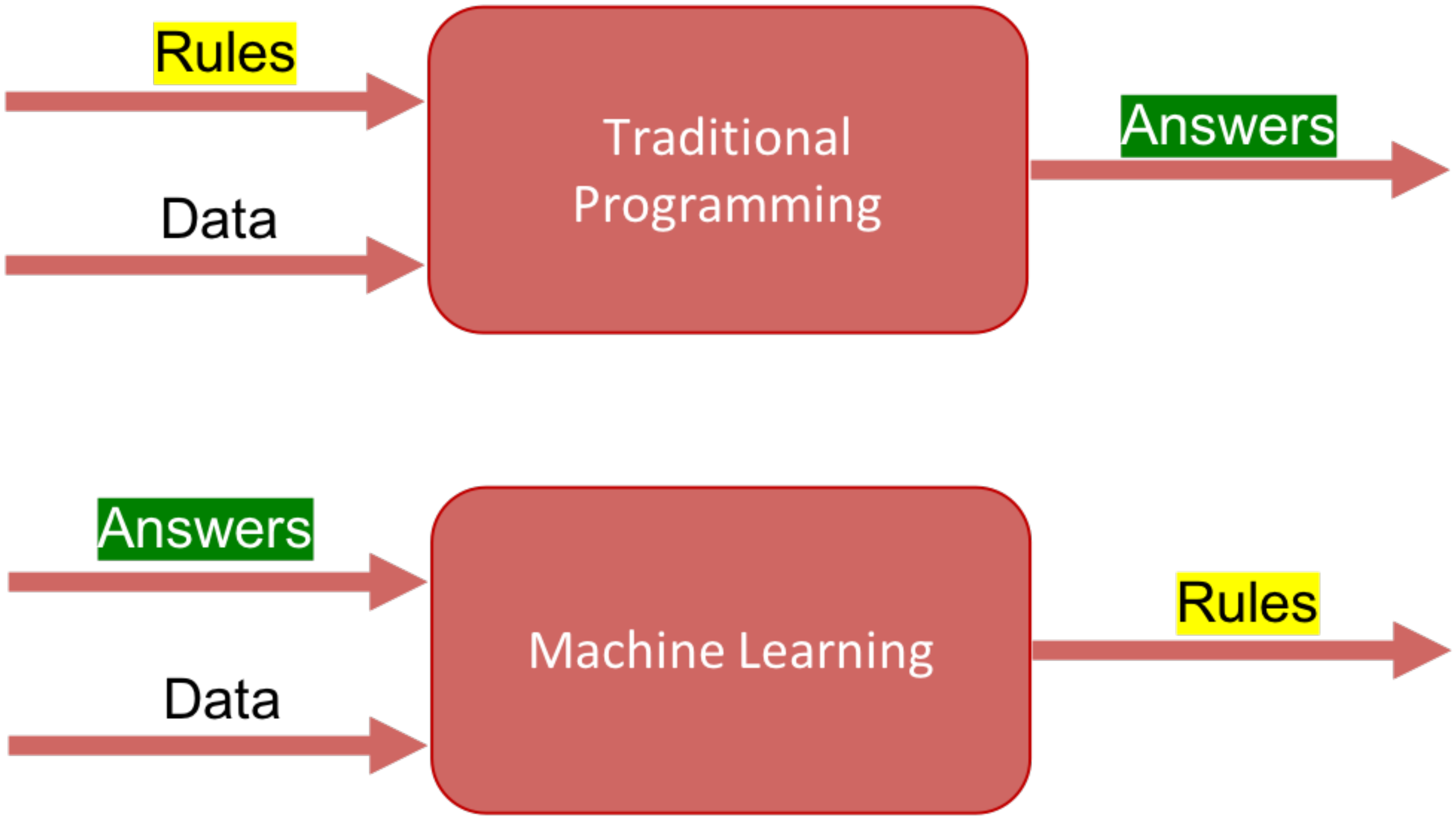
Traditional programming



Università degli Studi di Napoli Parthenope
Foundation Program

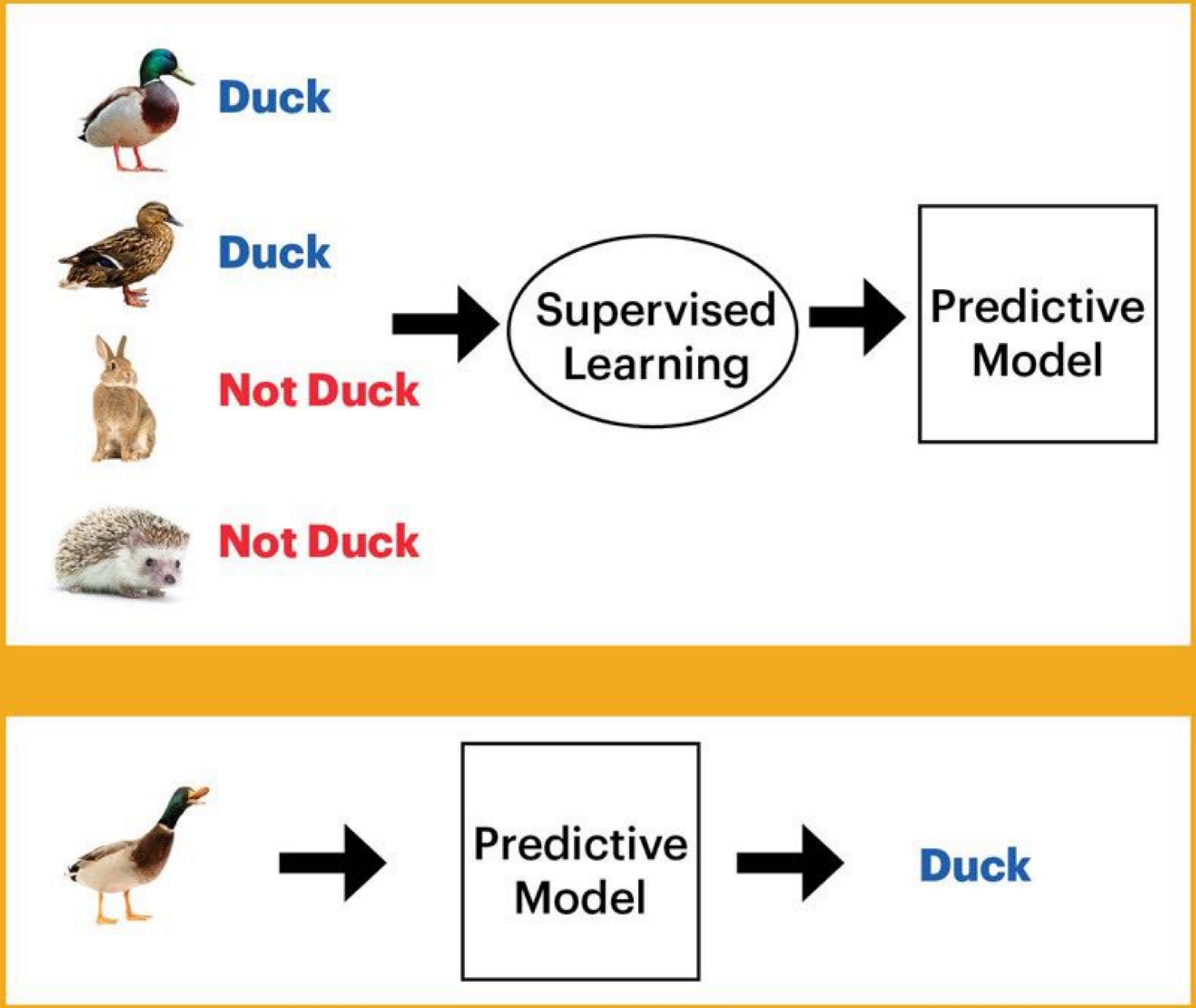
```
if (ball.collide(brick)){
    removeBrick();
    ball.dx=-1*(ball.dx);
    ball.dy=-1*(ball.dy);
}
```


Rules
Data
Traditional Programming
Answers

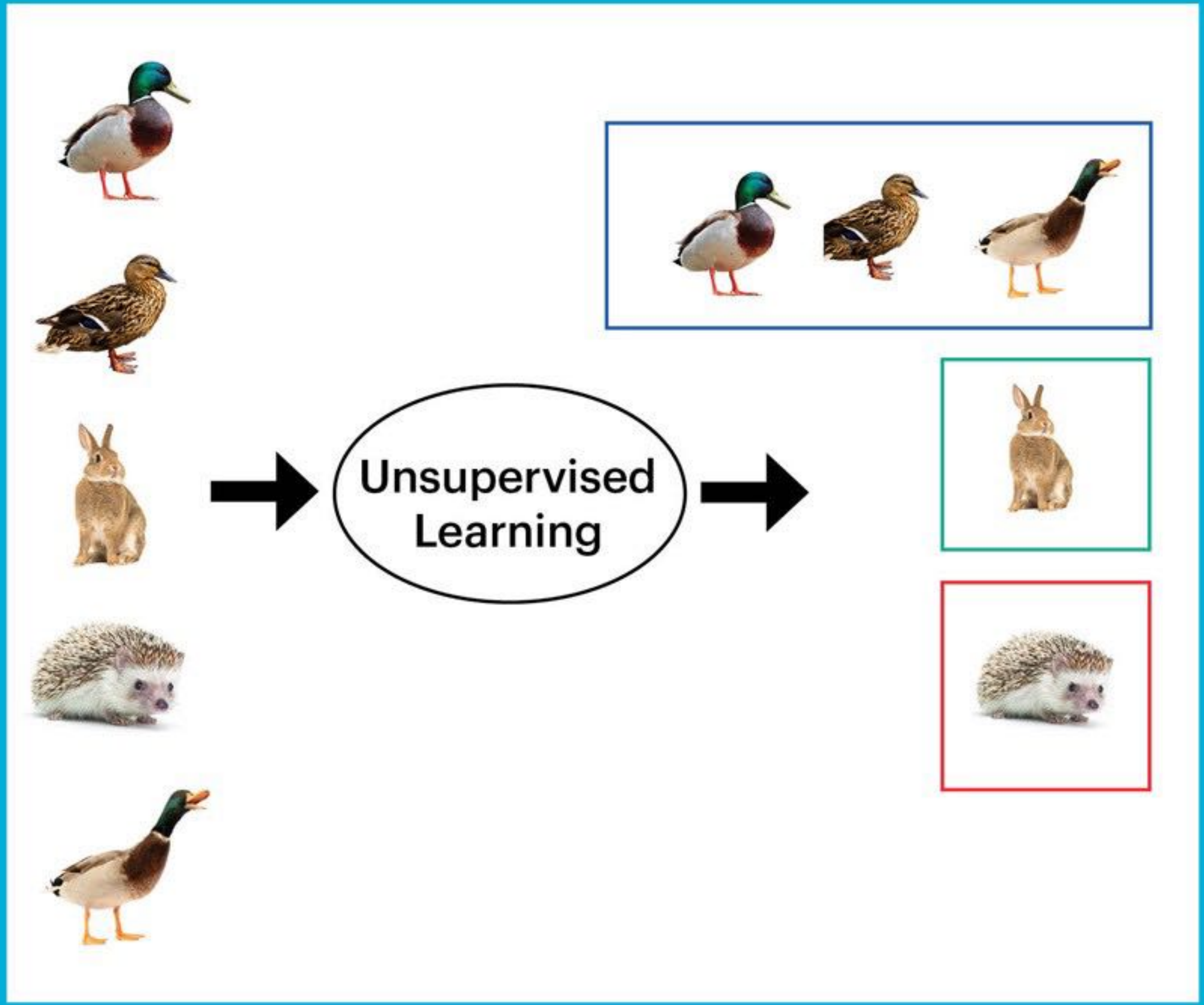

Rules
Data
Traditional Programming
Answers
Answers
Data
Machine Learning
Rules

Two general types of machine learning algorithms

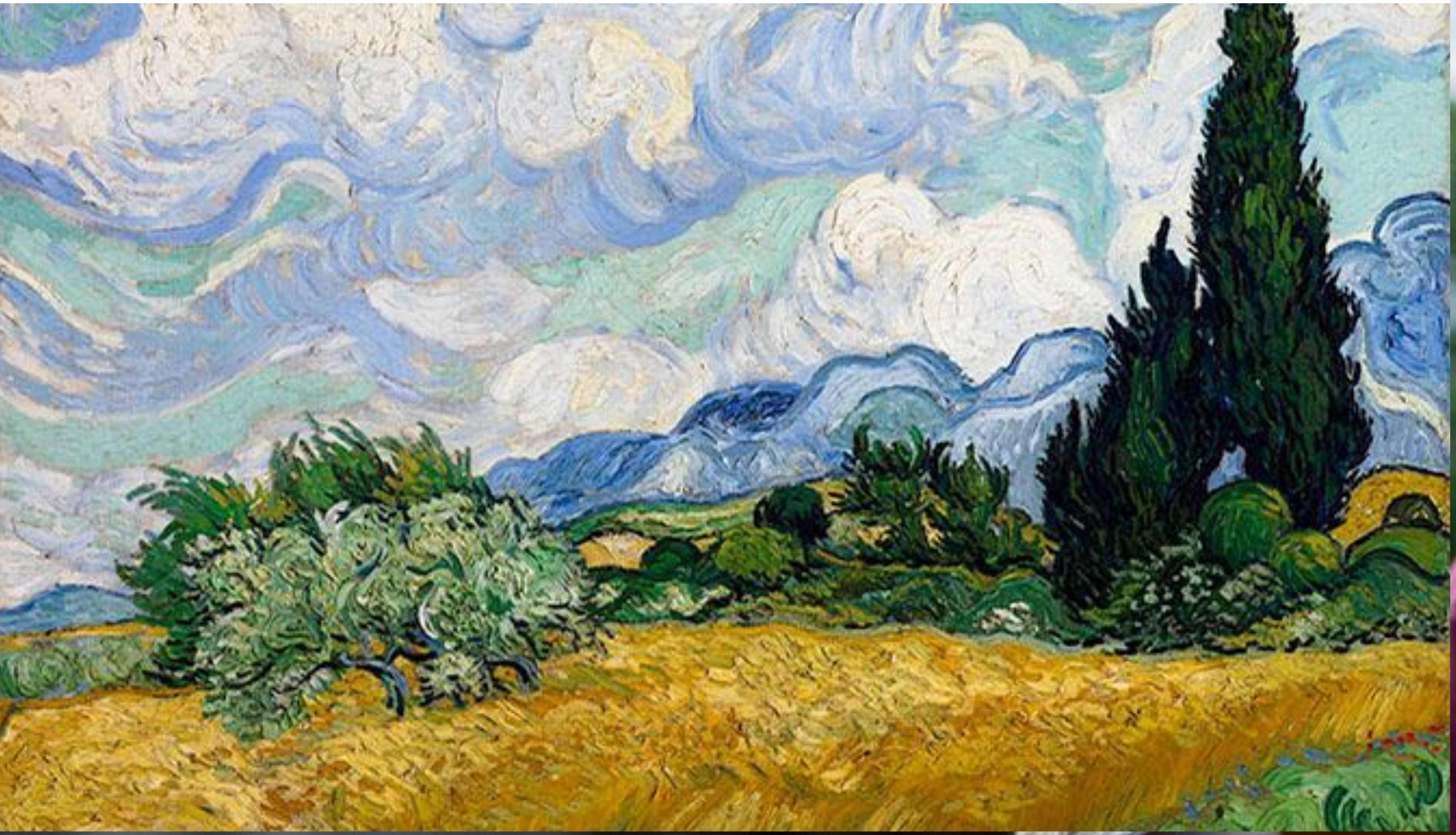
Supervised Learning (Classification Algorithm)



Unsupervised Learning (Clustering Algorithm)

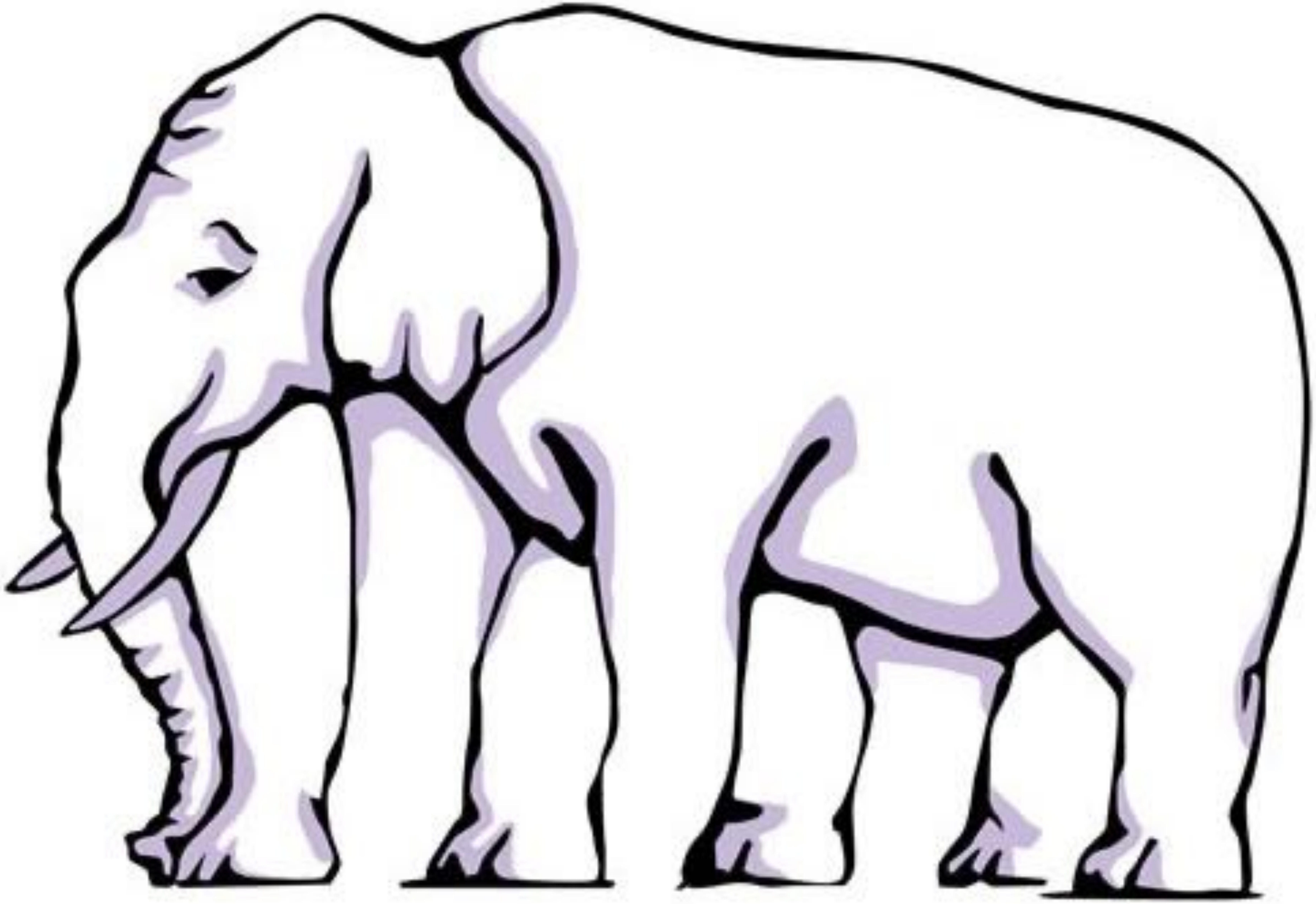


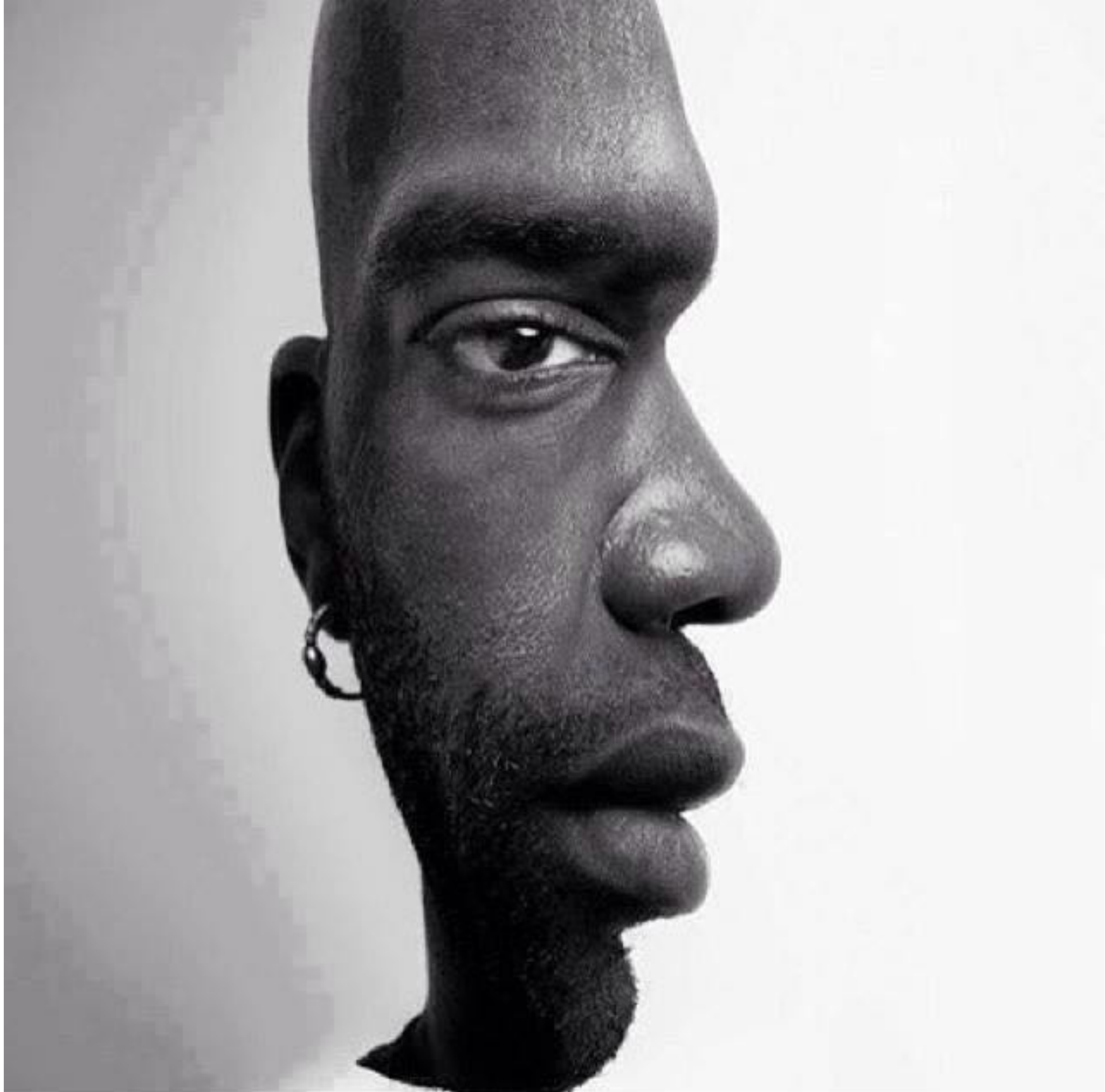
Let’s consider a specific type of data…

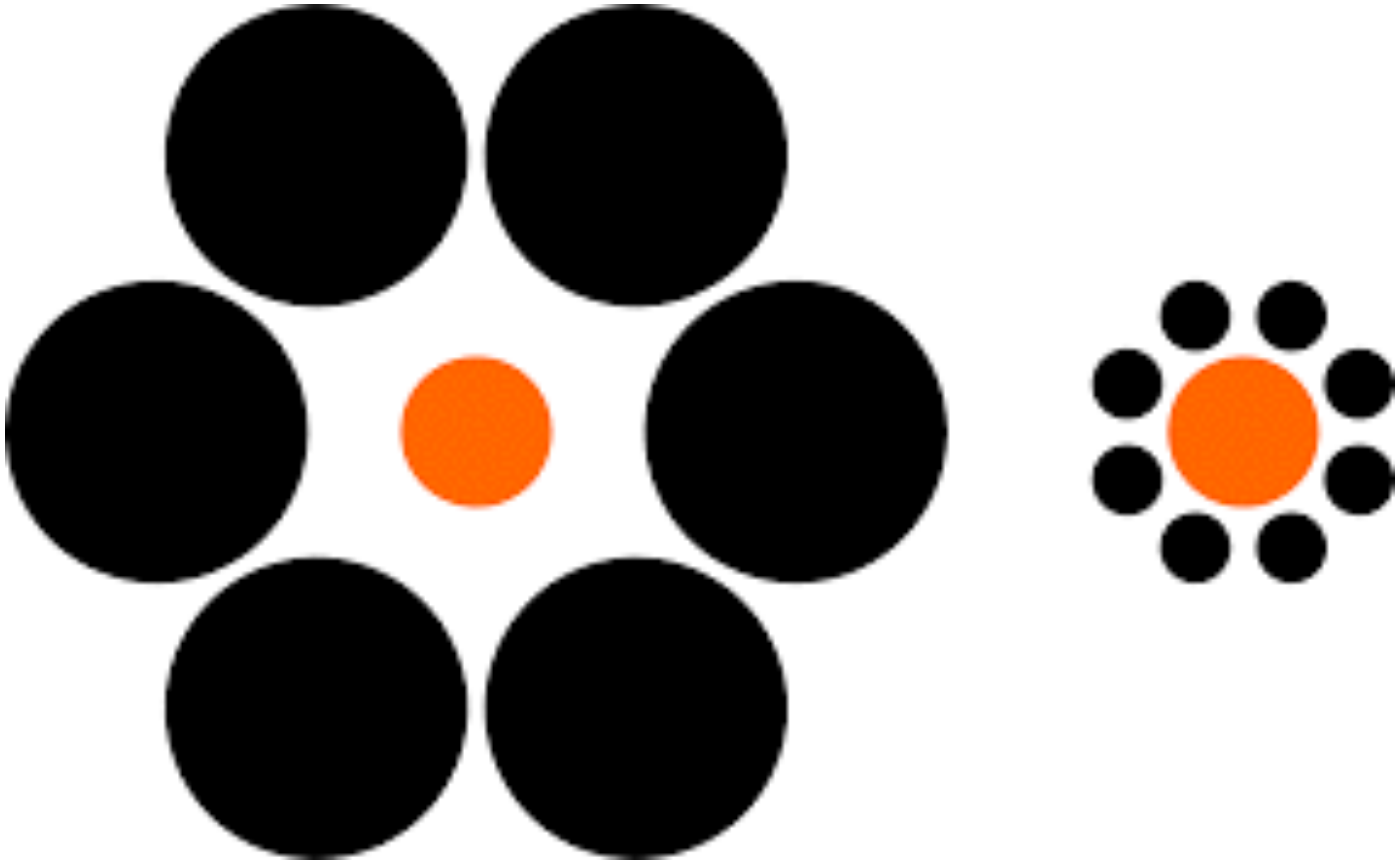


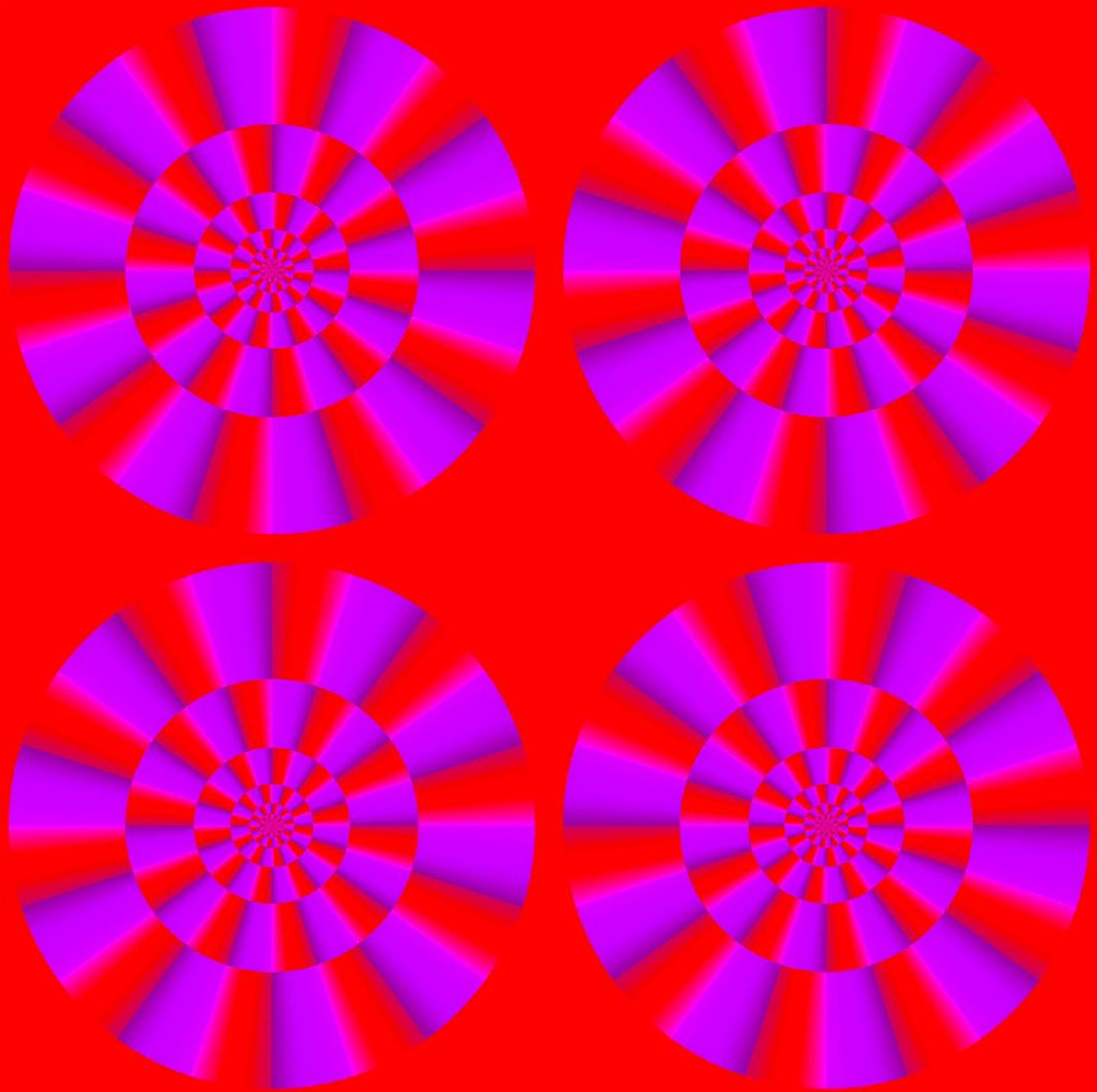

Given an **image**, can a machine predict what is there in that image?

Are human eyes foolproof?









Human eye is wonderful and complex
… but not foolproof!

How do computers see an image?

For a computer, a **picture is nothing but a bunch of numbers**.
Hence, it **can't** easily understand the **semantics** of it as a human does.

```
[[  9   1  29  70 114  76   0   8   4   5   5   0 111 162   9   8  62  62]
 [  3   0  33  61 102 106  34   0   0   0   0  49 182 150   1  12  65  62]
 [  1   0  40  54 123  90  72  77  52  51  49 121 205  98   0  15  67  59]
 [  3   1  41  57  74  54  96 181 220 170  90 149 208  56   0  16  69  59]
 [  6   1  32  36  47  81  85  90 176 206 140 171 186  22   3  15  72  63]
 [  4   1  31  39  66  71  71  97 147 214 203 190 198  22   6  17  73  65]
 [  2   3  15  30  52  57  68 123 161 197 207 200 179   8   8  18  73  66]
 [  2   2  17  37  34  40  78 103 148 187 205 225 165   1   8  19  76  68]
 [  2   3  20  44  37  34  35  26  78 156 214 145 200  38   2  21  78  69]
 [  2   2  20  34  21  43  70  21  43 139 205  93 211  70   0  23  78  72]
 [  3   4  16  24  14  21 102 175 120 130 226 212 236  75   0  25  78  72]
 [  6   5  13  21  28  28  97 216 184  90 196 255 255  84   4  24  79  74]
 [  6   5  15  25  30  39  63 105 140  66 113 252 251  74   4  28  79  75]
 [  5   5  16  32  38  57  69  85  93 120 128 251 255 154  19  26  80  76]
 [  6   5  20  42  55  62  66  76  86 104 148 242 254 241  83  26  80  77]
 [  2   3  20  38  55  64  69  80  78 109 195 247 252 255 172  40  78  77]
 [ 10   8  23  34  44  64  88 104 119 173 234 247 253 254 227  66  74  74]
 [ 32   6  24  37  45  63  85 114 154 196 226 245 251 252 250 112  66  71]]
```

…let me ask you to write a **traditional algorithm… to solve a (simple) problem**

Are you able to write some RULES to detect a cat from an image?

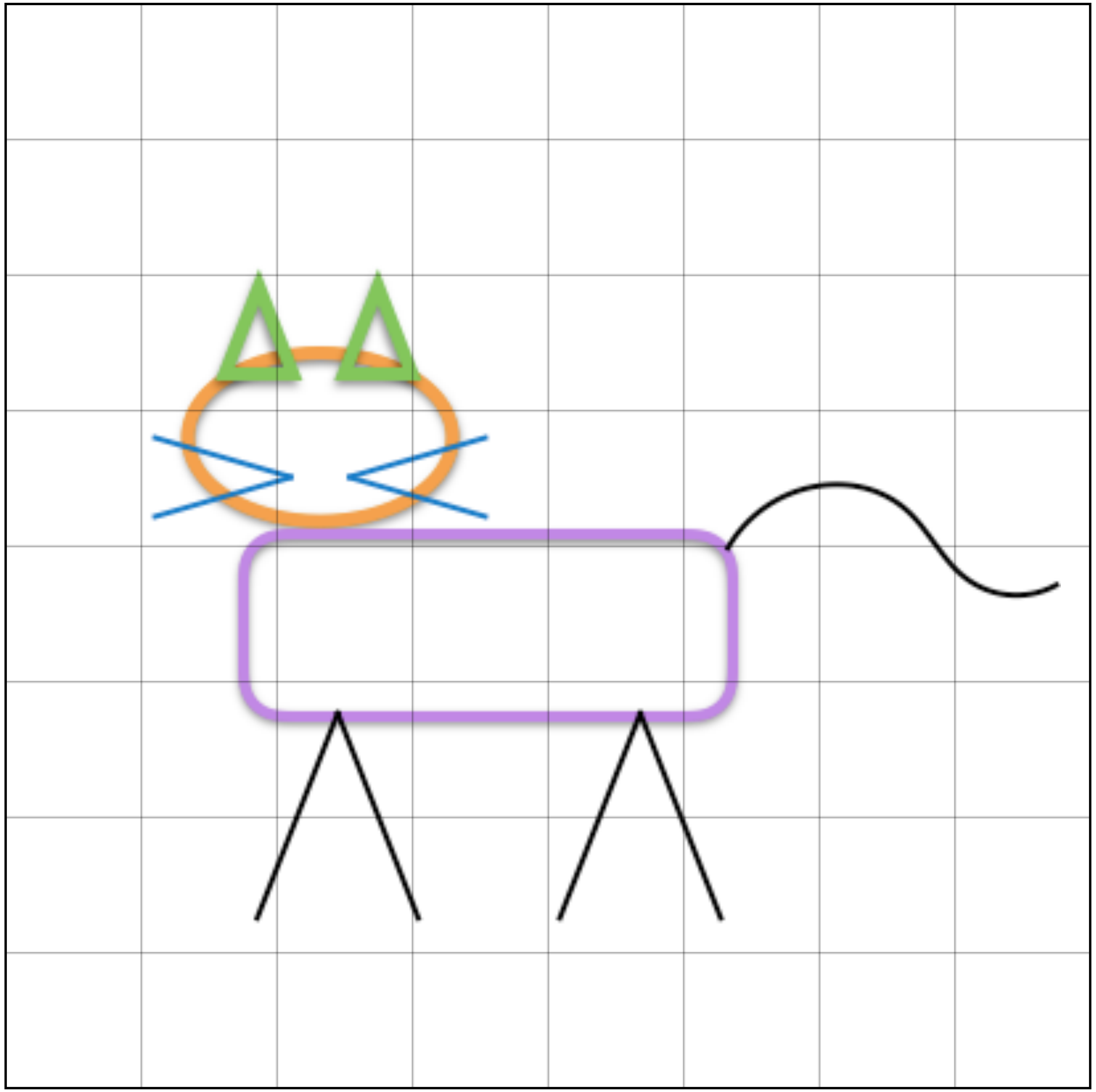
Let’s do some basic assumptions on the cat:

- 2 **ears**
- an oval **face** with **whiskers**
- a cylindrical **body**
- 4 **legs**
- a curvy **tail**

Assume we have written some **rules** to find **features** in an **image** which when combined form a cat that looks nearly as shown in the figure here.

Rule 1 . . .

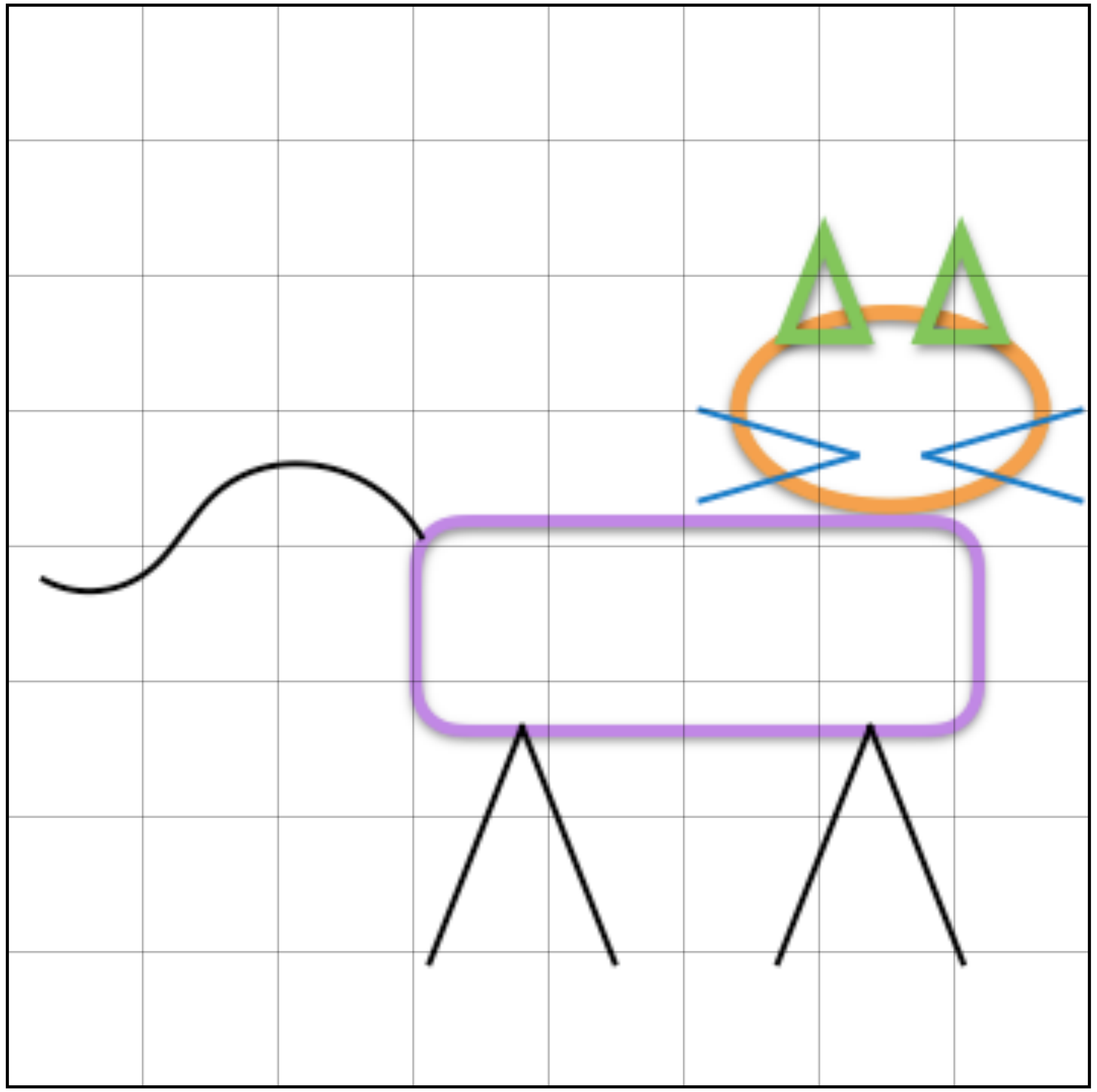
Rule 2 . . .



Let's test the performance on some real world images.

Can our algorithm accurately predict the cat in this picture?

- If you carefully observe the cat image with primitive shapes, we have actually some rules to find the cat that is turning towards only on **its left** ☹️.
- Write exact the same **reversed rules** for a cat turning towards its right 😎.
- Good! Now we have the **cat detector**!



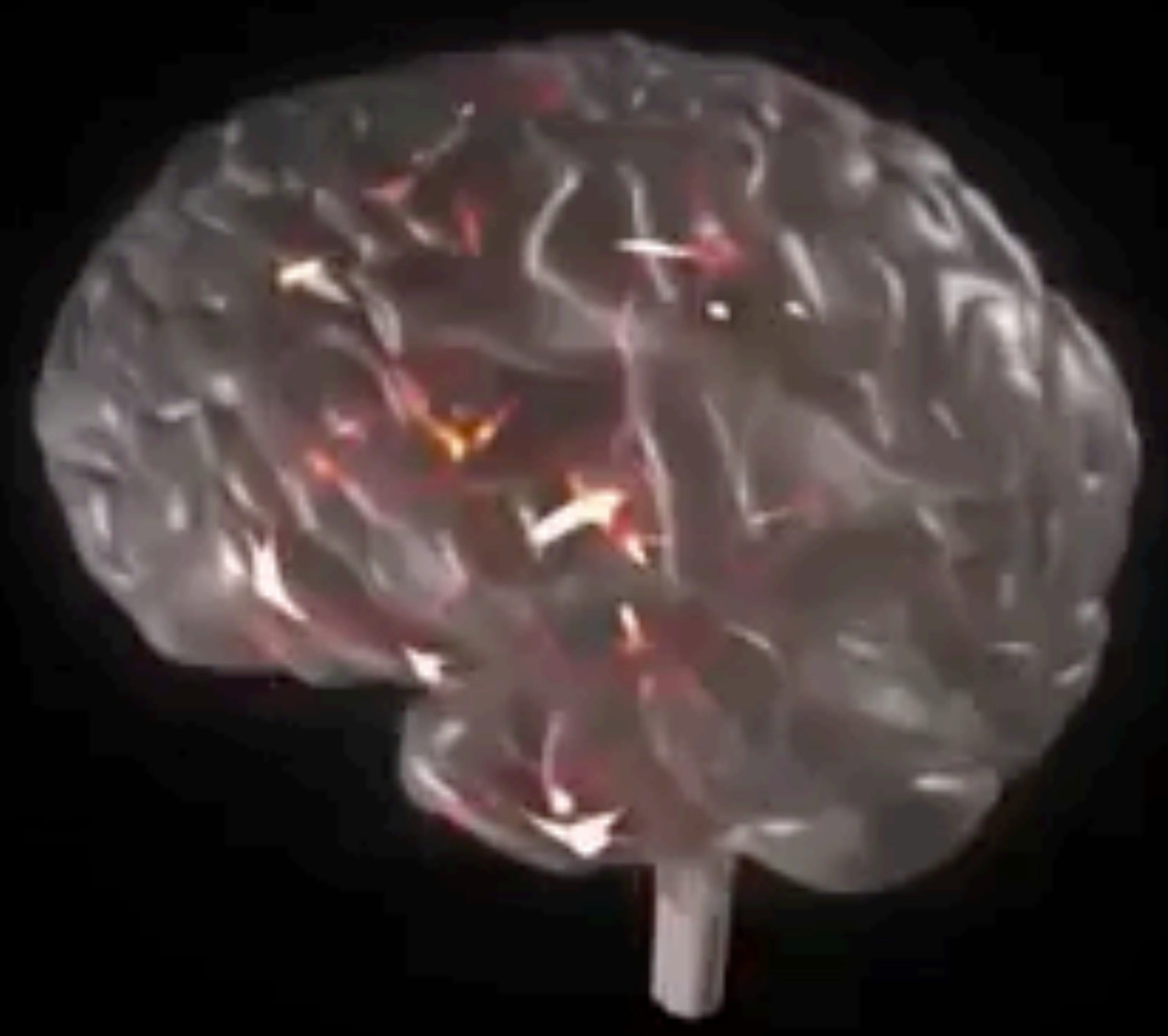
but cats are curious animals…

Can we detect the cat in this image?

We need another type of “rules”

Artificial Neural Networks

What do we know of our brain?



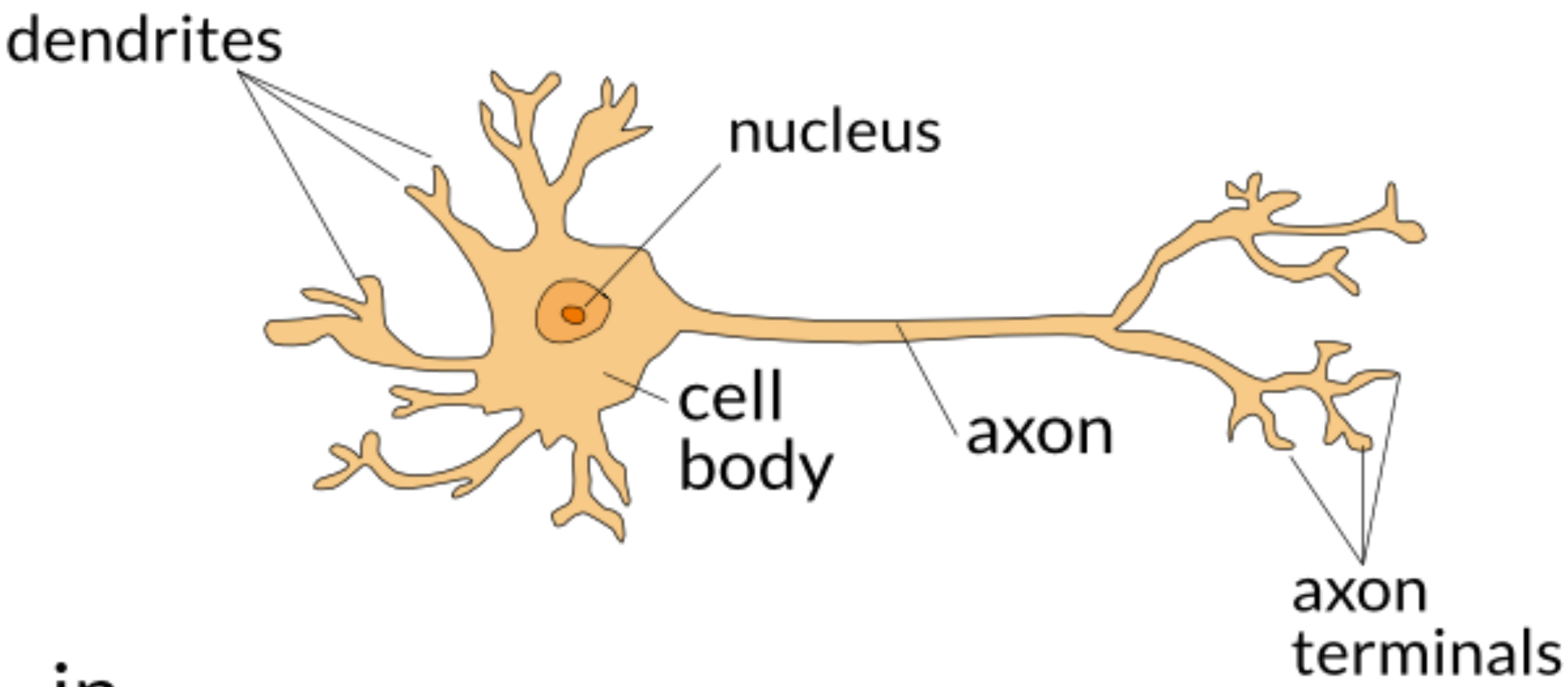
Mathematical model of an Artificial Neural Networks

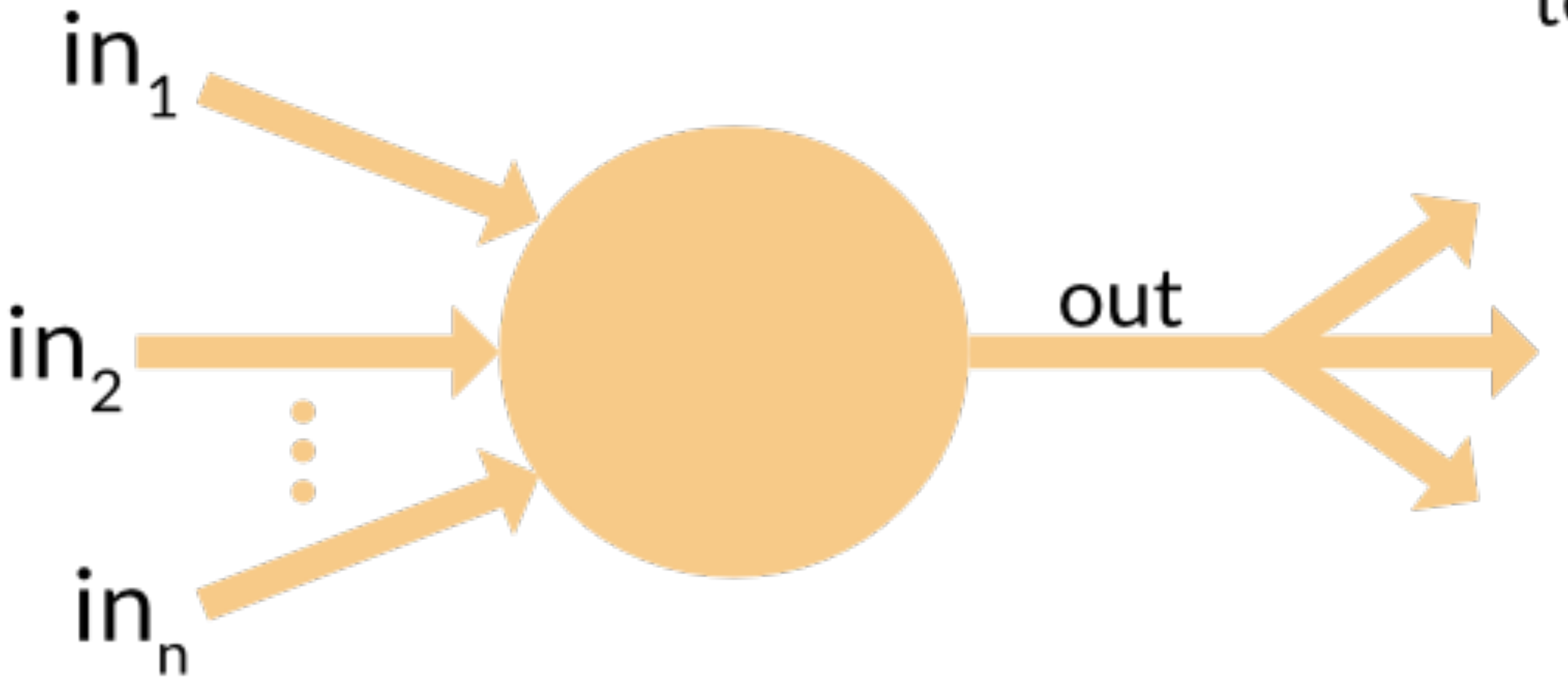
Biological inspiration…

One of the reasons for the origin of AI was to **emulate** the function of **neurons** in the human body.

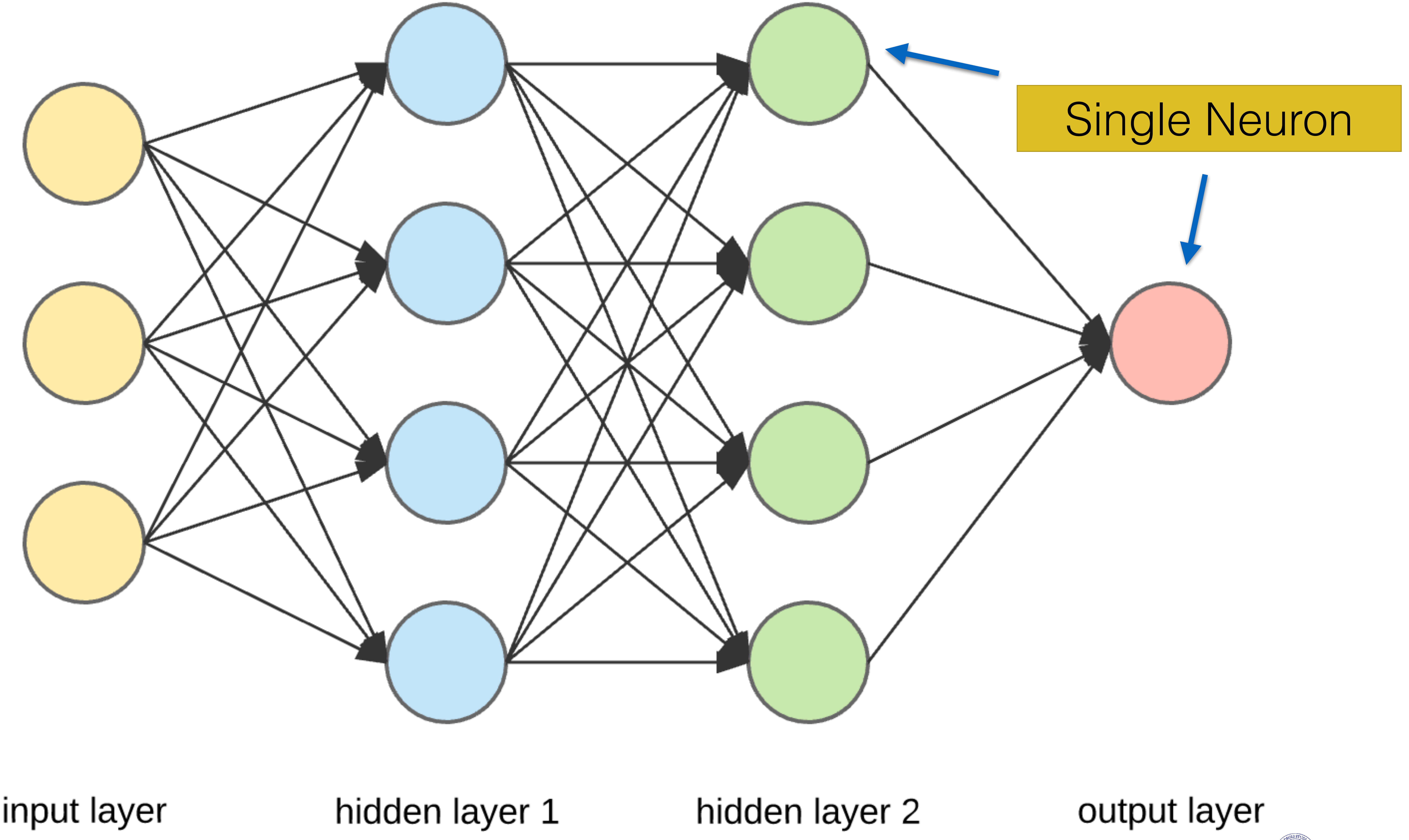
This way computers and machines can **imitate** nature's creation, the human **brain**, and perform tasks as fast and with as much accuracy as the human brain functions.

This is now done using what is called as **artificial neurons**.

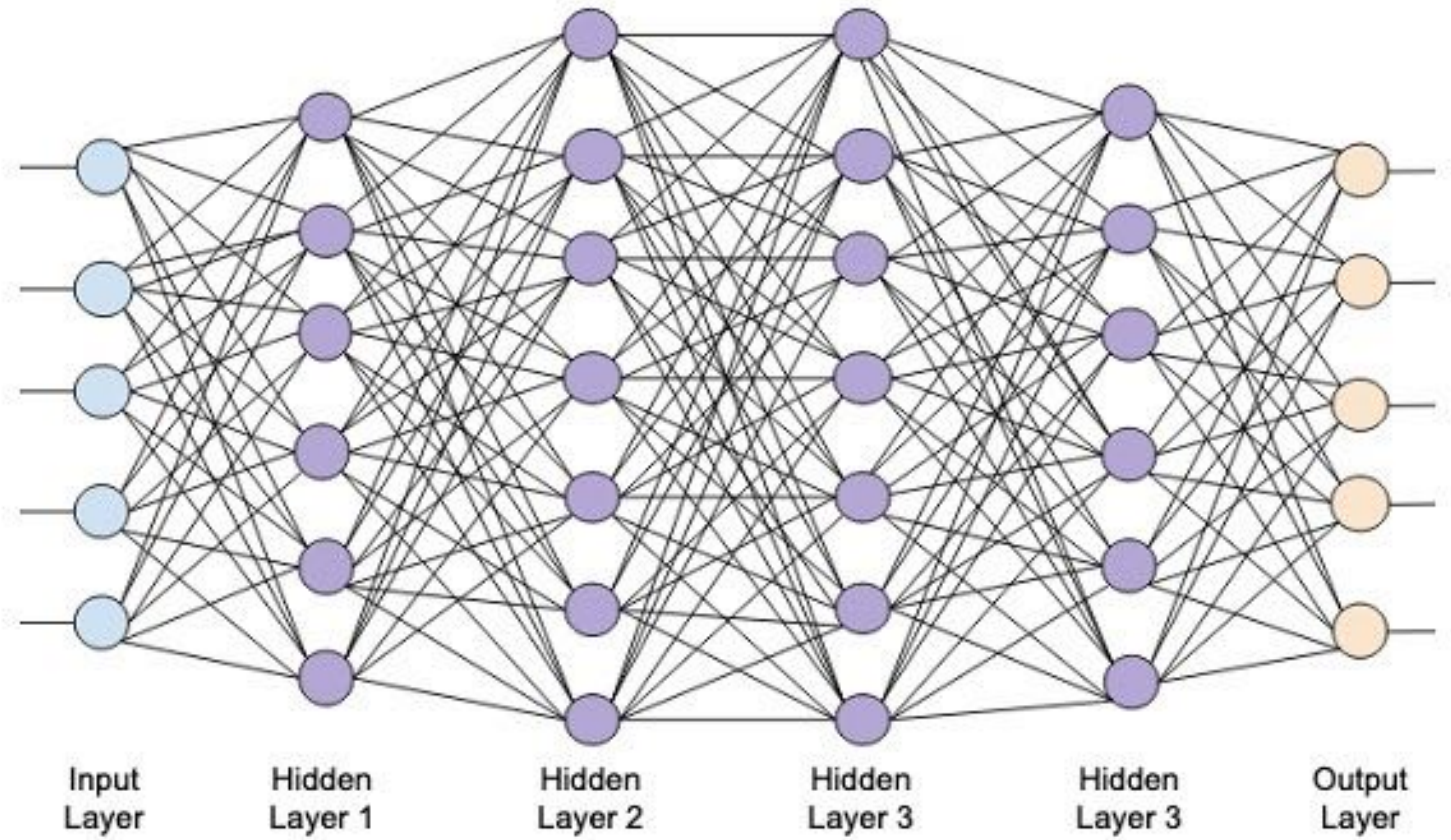




Artificial Neural Networks (ANN) are multi-layer fully-connected neural nets



More complex (deeper) model of ANN

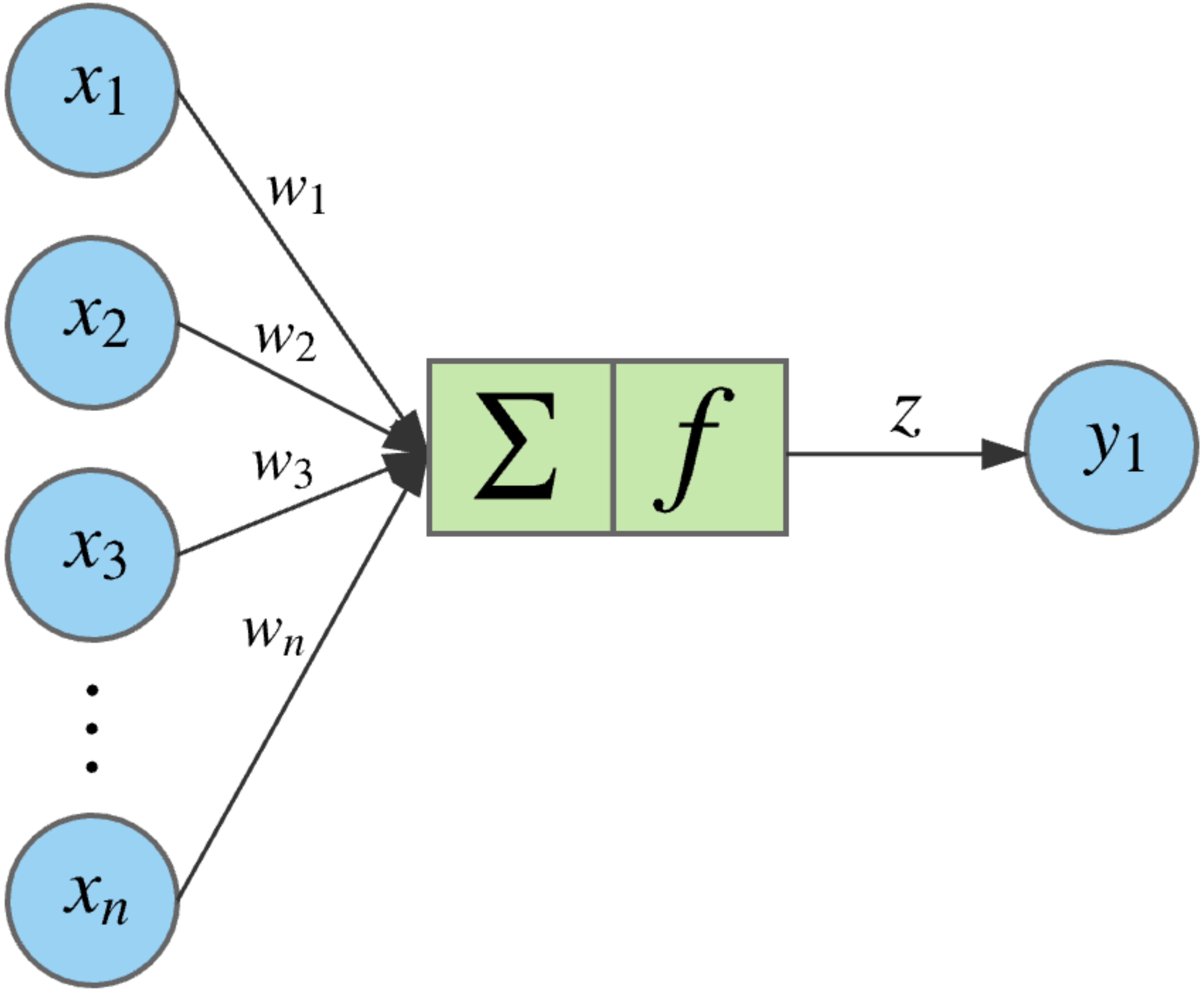


Artificial neuron model

A given node takes the weighted sum of its inputs, and passes it through a function.

This is the output of the node, which then becomes the input of another node in the next layer.

The signal flows from left to right, and the final output is calculated by performing this procedure for all the nodes.



The **equation for a given node** looks as follows.

The **weighted sum** of its inputs passed through **a non-linear activation function**. It can be represented as a vector dot product, where ***n*** is the number of inputs for the node.

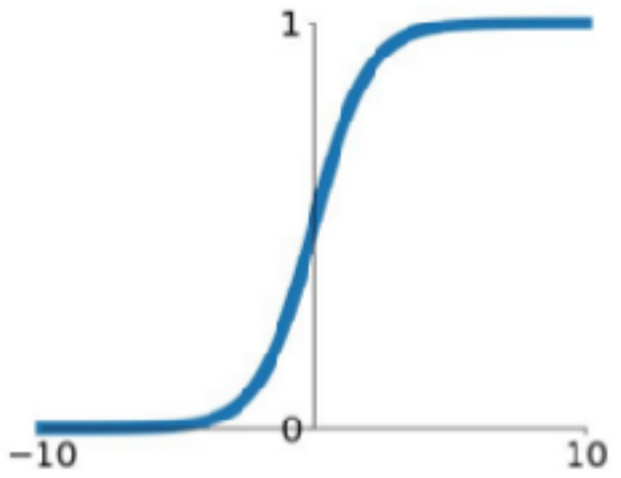
$$z = f(x \cdot w) = f\left(\sum_{i=1}^{n} x_i w_i\right)$$

$$x \in d_{1\times n},\ w \in d_{n\times 1},\ z \in d_{1\times 1}$$

f is a non linear “activation” function $\longrightarrow$ $f\left(\sum_{i=1}^{n} x_i w_i\right)$

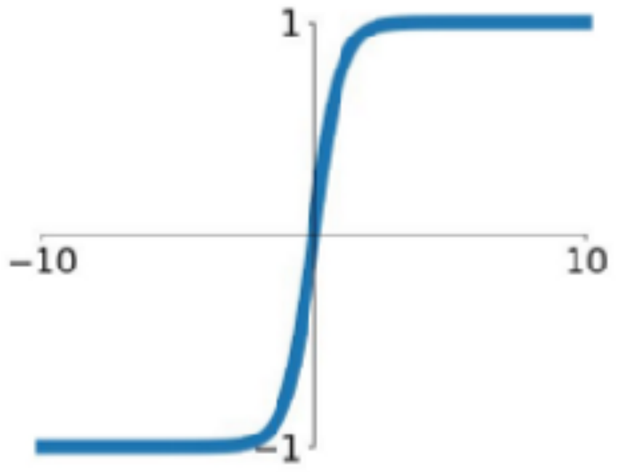
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



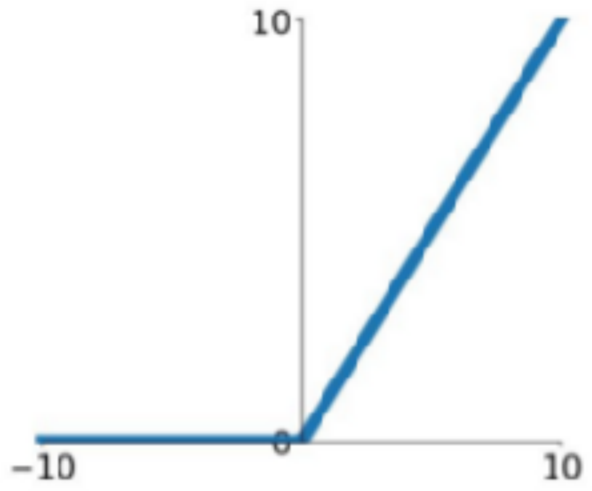
tanh

$\tanh(x)$



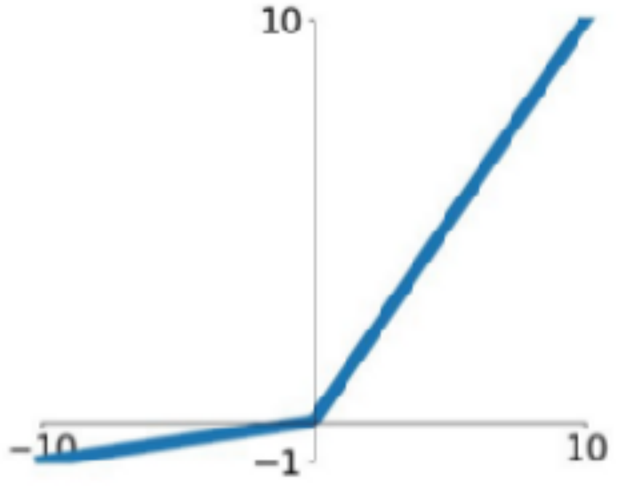
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

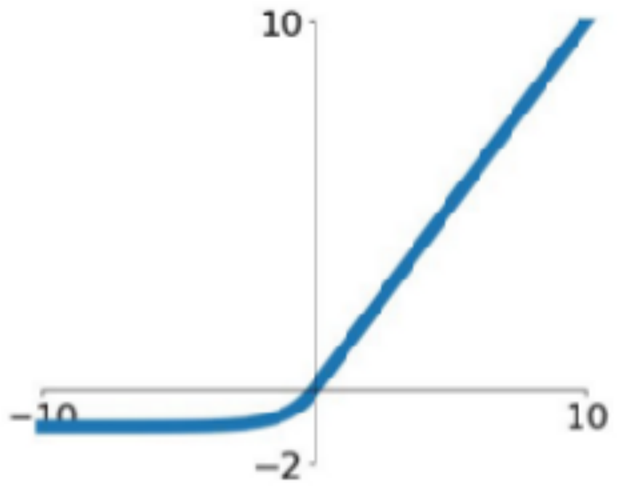


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

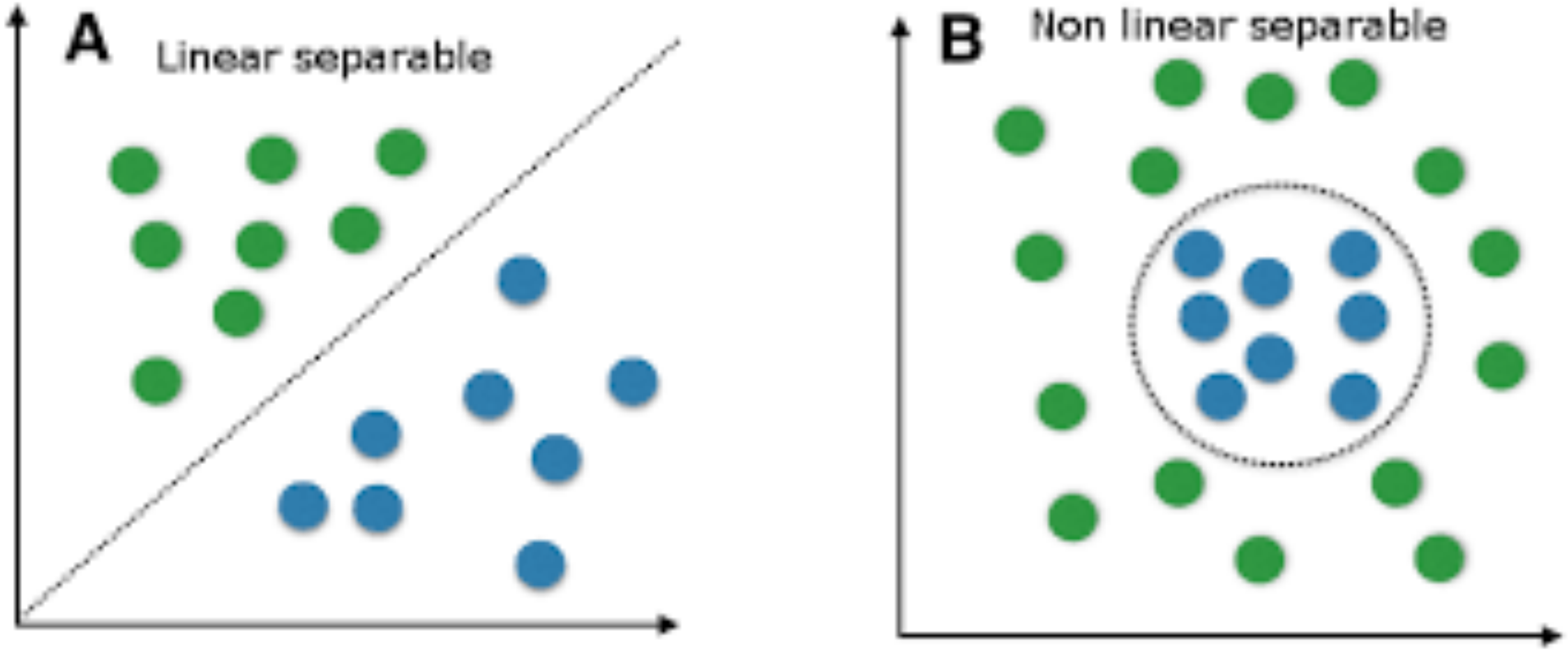


Some common activation functions

The purpouse of the activation functions is to introduce non-linearity inside the network

- For some tasks, input data can be linearly separable, and linear classifiers can be suitably applied

- For other tasks, linear classifiers may have difficulties to produce adequate decision boundaries



The training algorithm

1. Randomly **initialize the weights** for all the nodes.

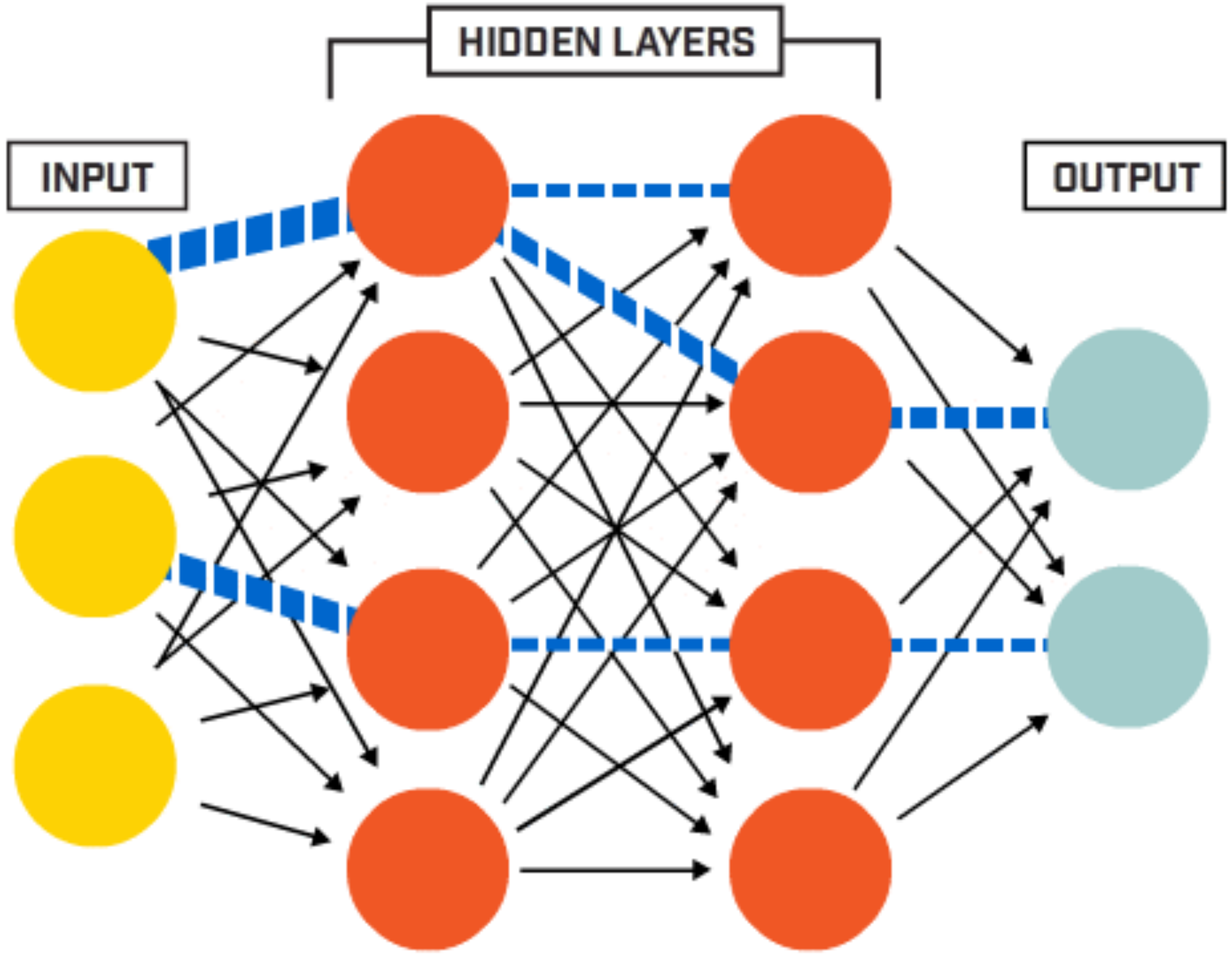
2. For every training example, **perform a forward pass** using the current weights, and calculate the output of each node going from left to right. The final output is the value of the last node.

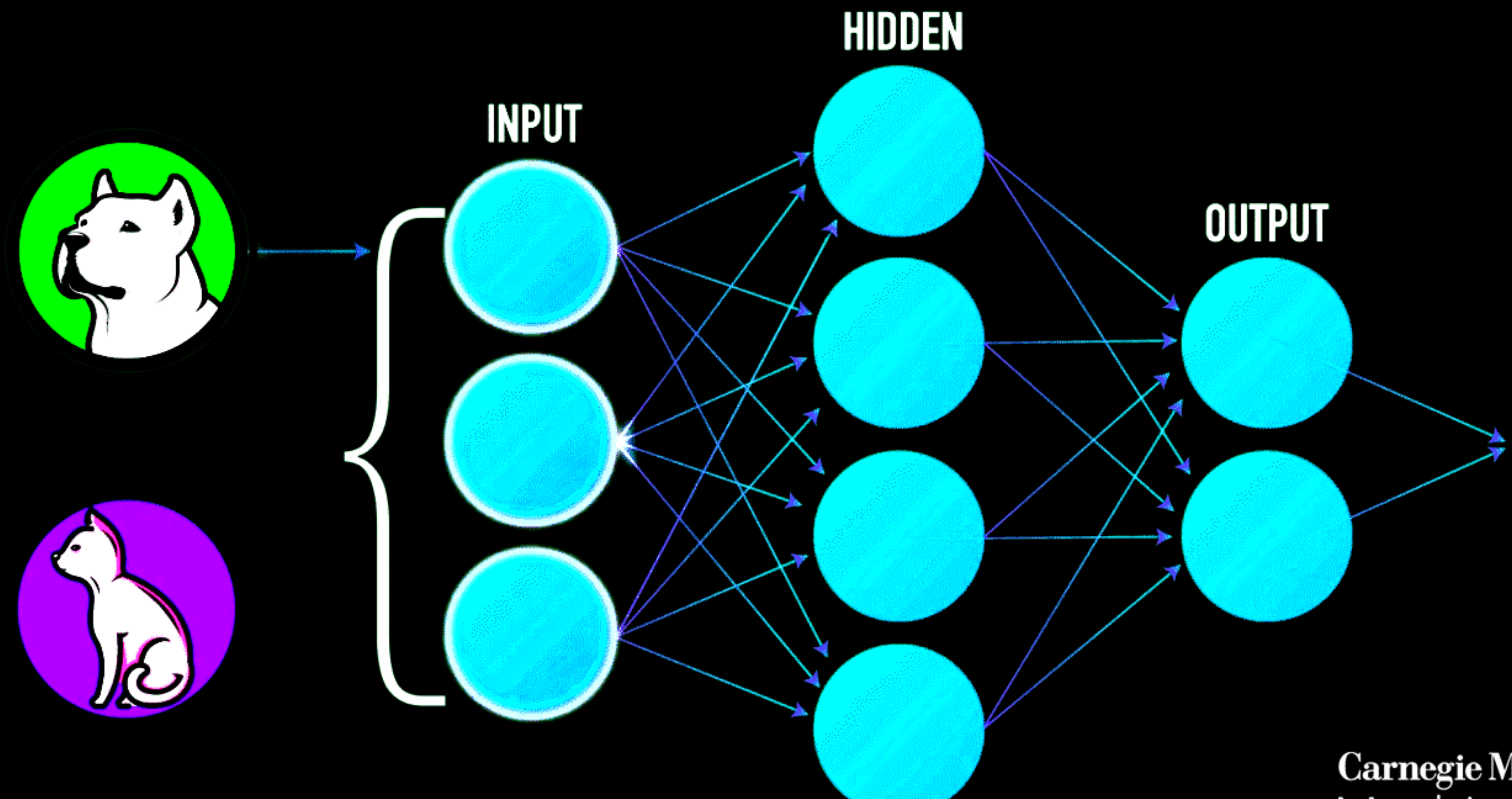
3. Compare the final output with the actual target in the training data, and measure the error using a ***loss function***.

4. Perform **a *backwards pass*** from right to left and propagate the error to every individual node using ***backpropagation***. Calculate each weight's contribution to the error, and adjust the weights accordingly using ***gradient descent***. Propagate the error gradients back starting from the last layer.

ANN (forward pass)

Only some signals (**values**)
are propagated
through the networks
(due to the **weights** and **bias**)





Carnegie Mellon University
Machine Learning

We need a function to calculate our error.
This is also called **cost function** or **loss function**

There are many available loss functions, the nature of our problem should dictate our choice of loss function. Here we'll use a simple **sum-of-squares error** as our loss function.

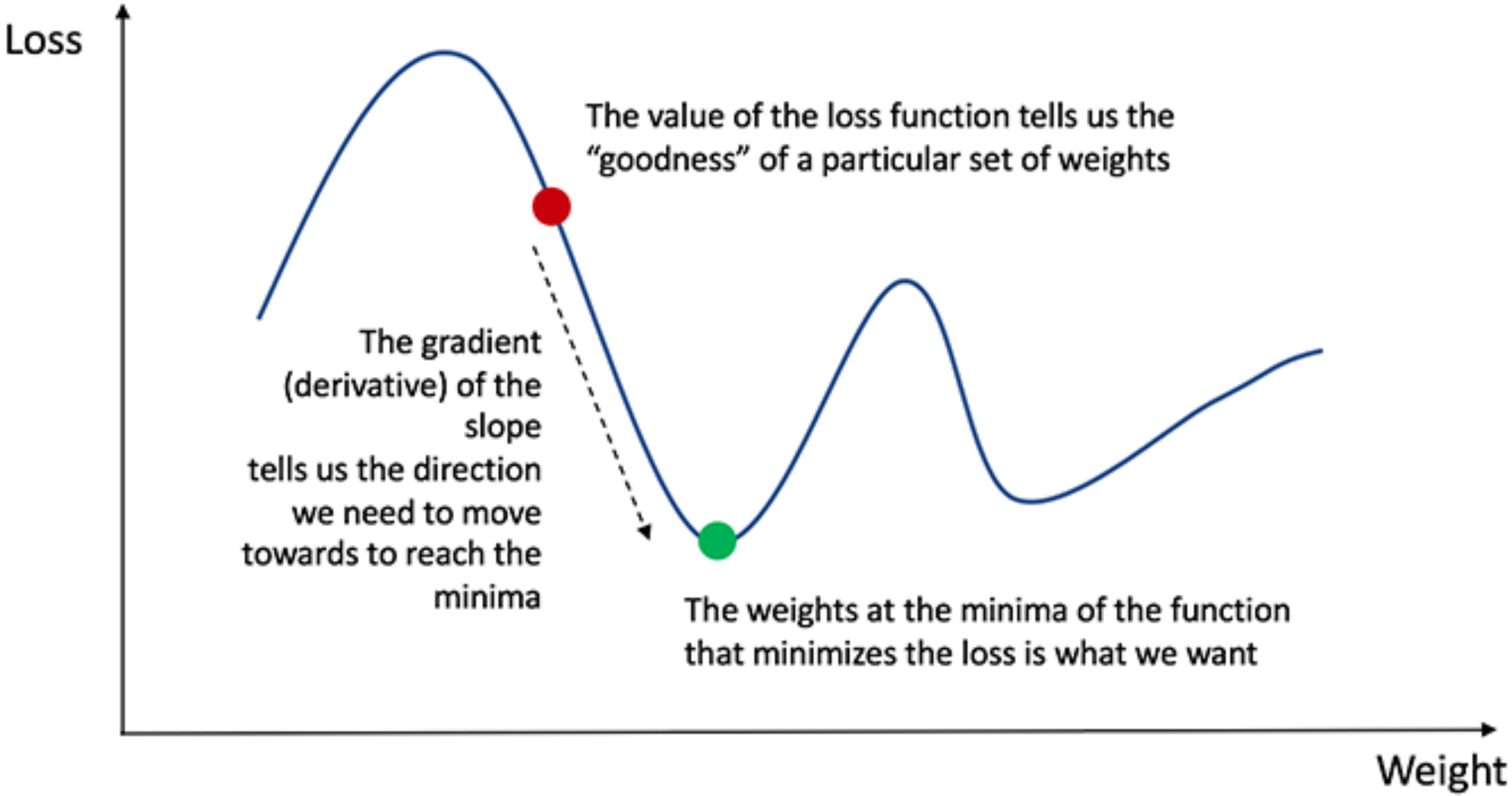
Mean Sum of Squares Error = $\dfrac{1}{2}\displaystyle\sum_{i=1}^{n}(out - y)^2$

out predicted output

y desired output

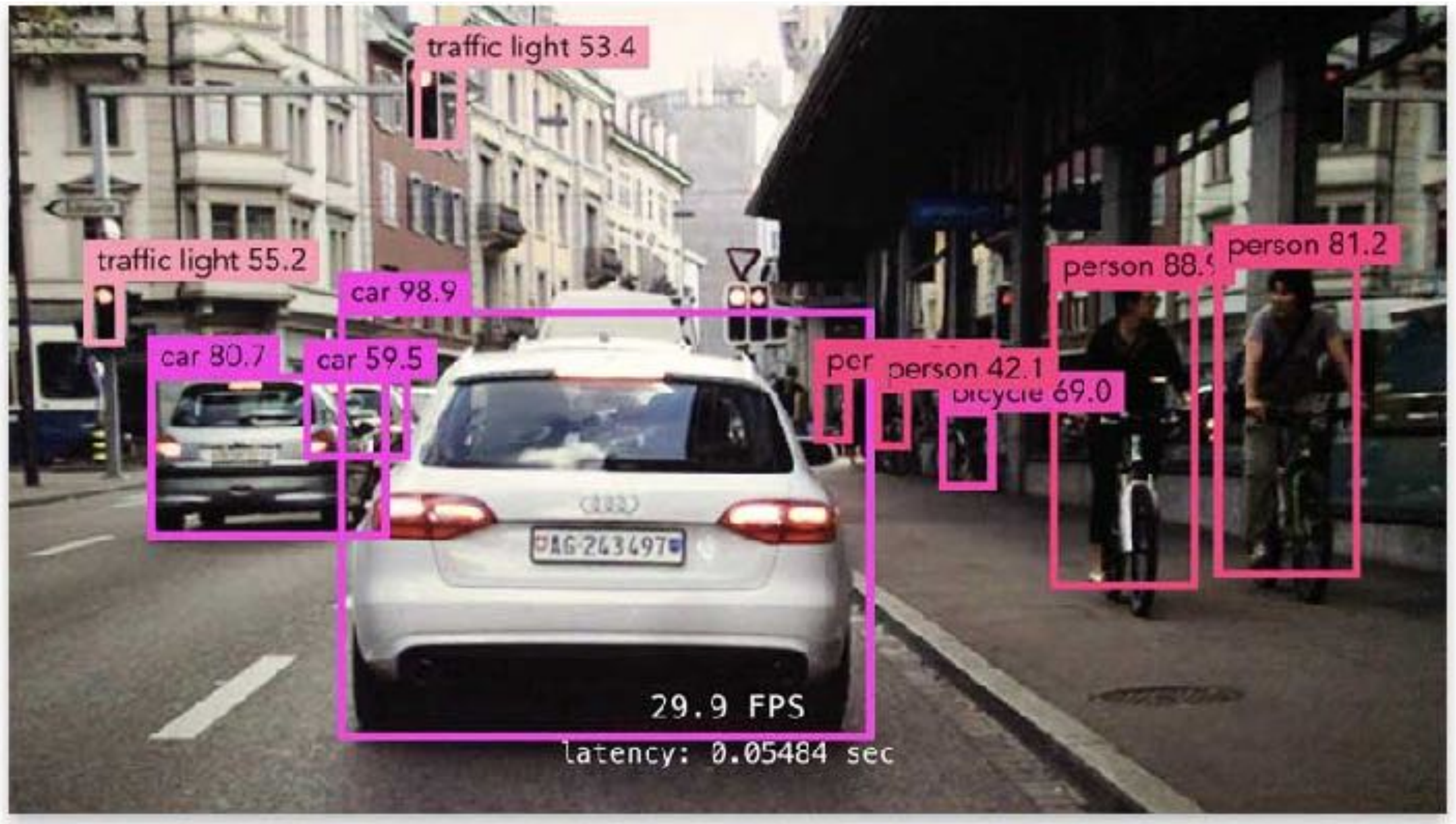
The sum-of-squares error is simply the sum of the difference between each predicted value and the actual value. The difference is squared so that we measure the absolute value of the difference.

Our goal in training is to find the best set of weights that minimizes the loss function.

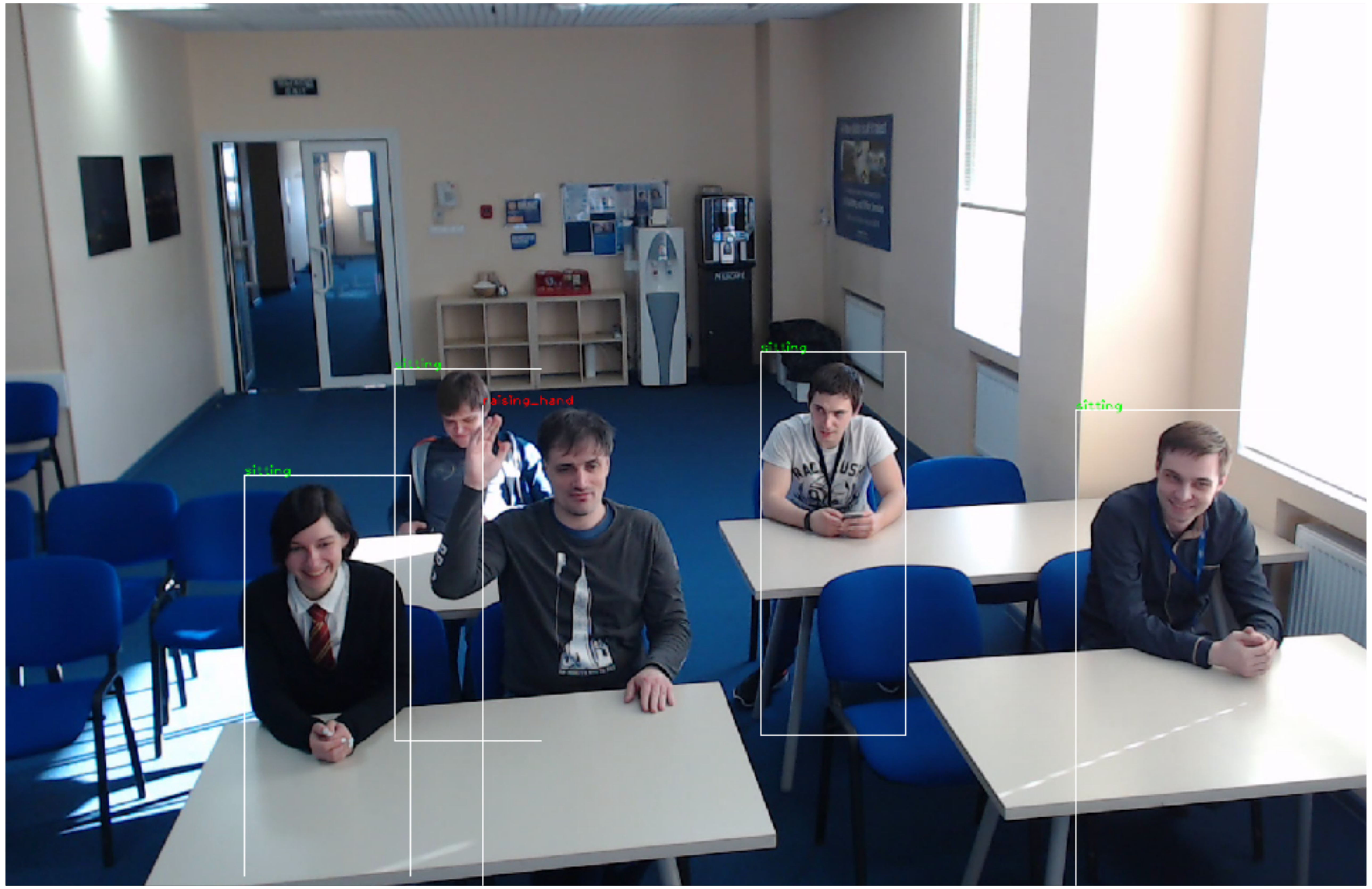

Loss
The value of the loss function tells us the "goodness" of a particular set of weights
The gradient (derivative) of the slope tells us the direction we need to move towards to reach the minima
The weights at the minima of the function that minimizes the loss is what we want
Weight

Convolutional Neural Networks
(CNN)

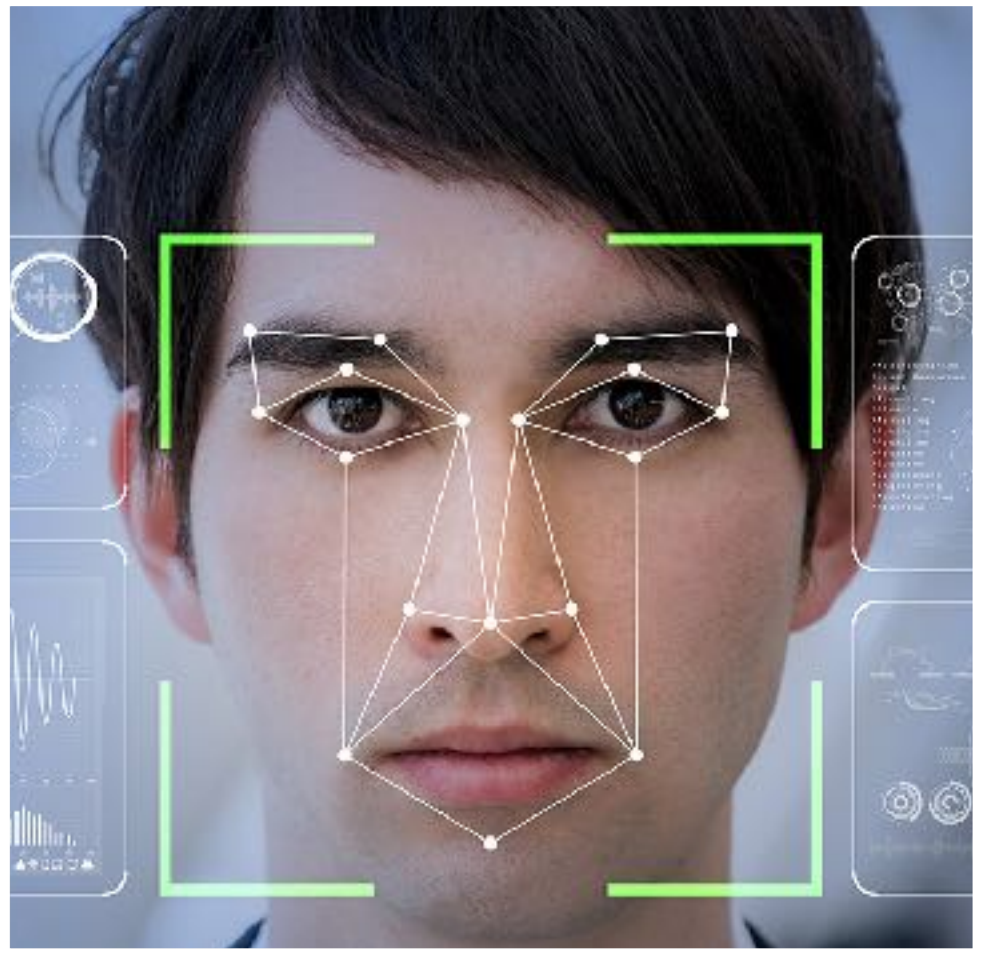
What CNNs can do?



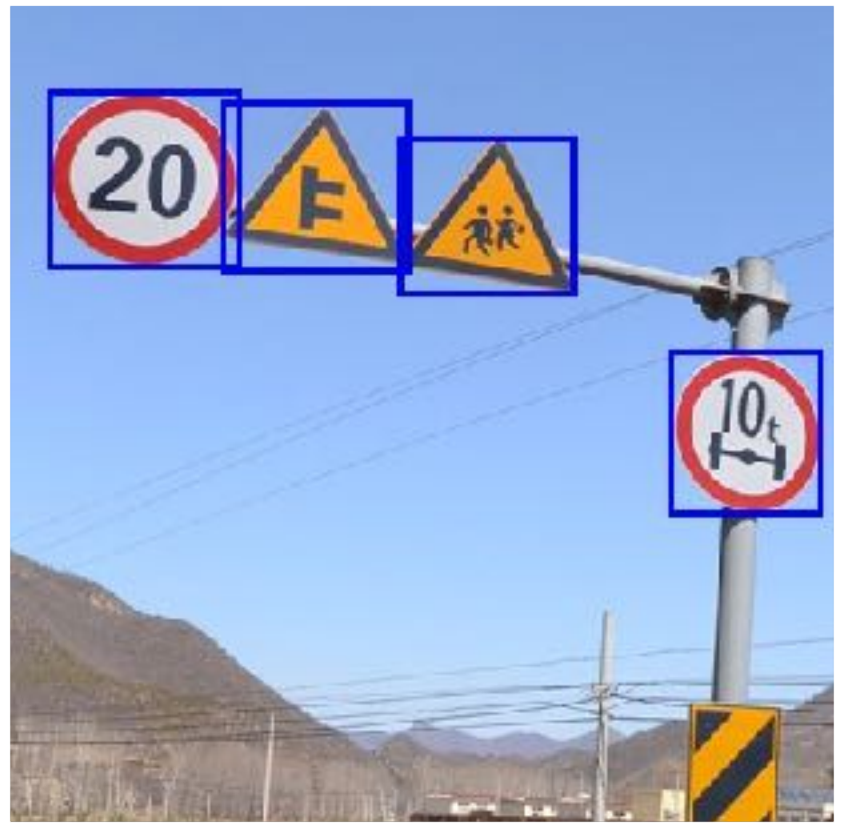
Objects detection



Action recognition



Face detection



Signs detection

CNN derive their name from the “**convolution**” operator.
The primary purpose of Convolution in case of a CNN is to **extract features** from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data.

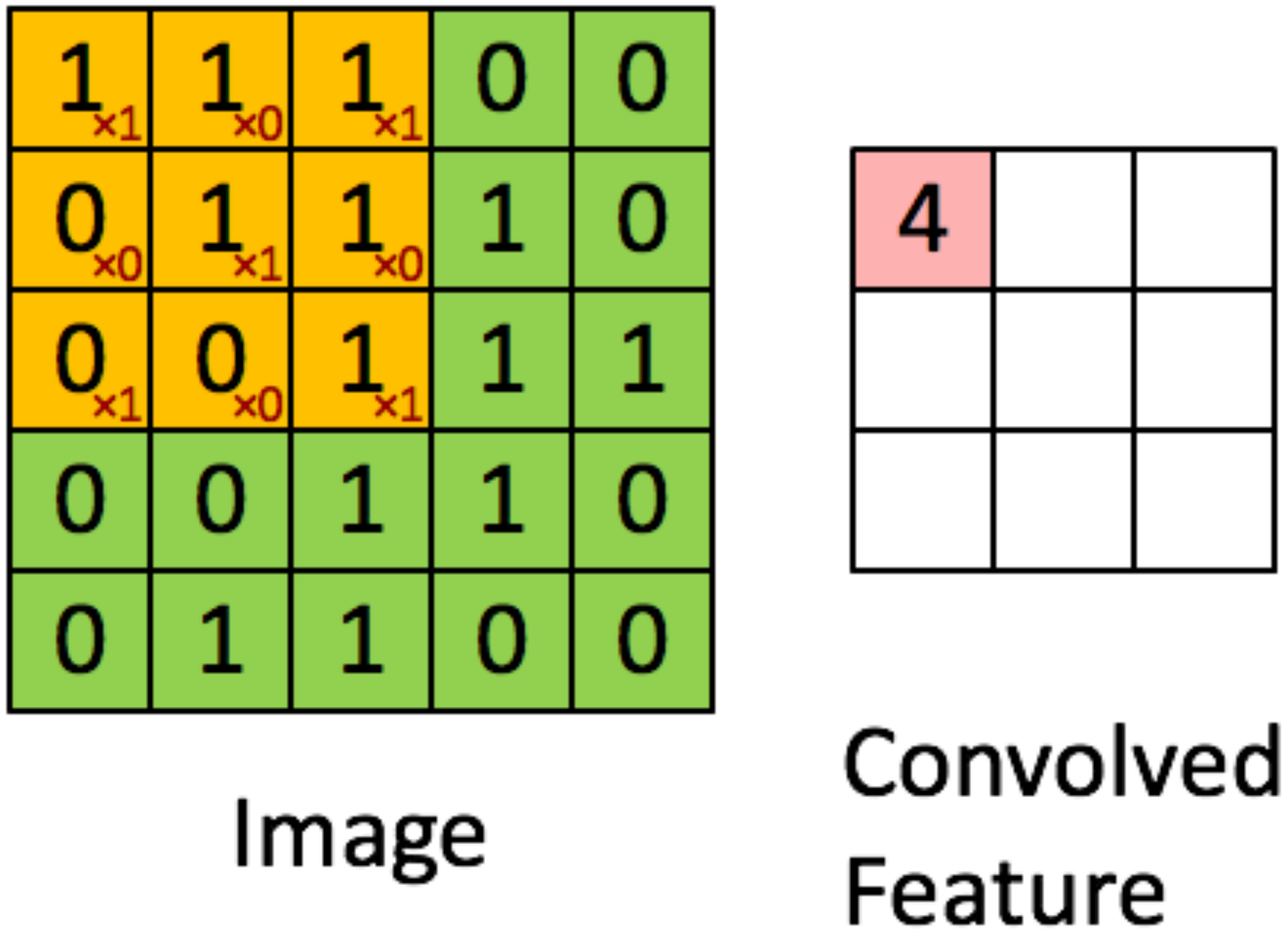
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Consider a 5 x 5 image whose pixel values are only 0 and 1 (note that for a grayscale image, pixel values range from 0 to 255, the green matrix below is a special case where pixel values are only 0 and 1)

1	0	1
0	1	0
1	0	1

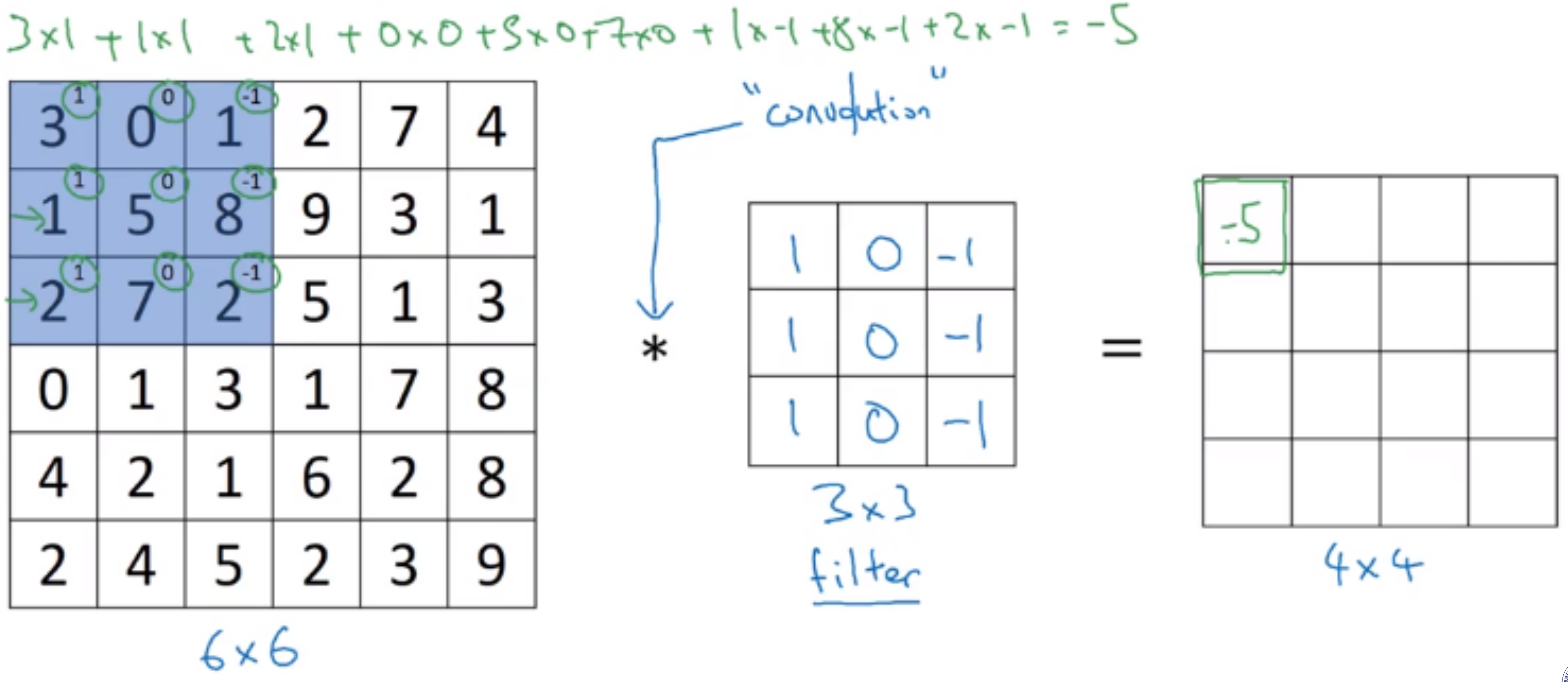
Also, consider another 3 x 3 matrix

Then, the **convolution** of the 5 x 5 image and the 3 x 3 matrix can be computed as shown in this animation



In CNN terminology, the 3×3 matrix is called a ‘**filter**‘ or ‘kernel’ or ‘feature detector’ and the matrix formed by sliding the filter over the image and computing the dot product is called the‘Activation Map’ or the ‘**Feature Map**‘.

It is important to **note that filters acts as feature detectors** from the original input image.

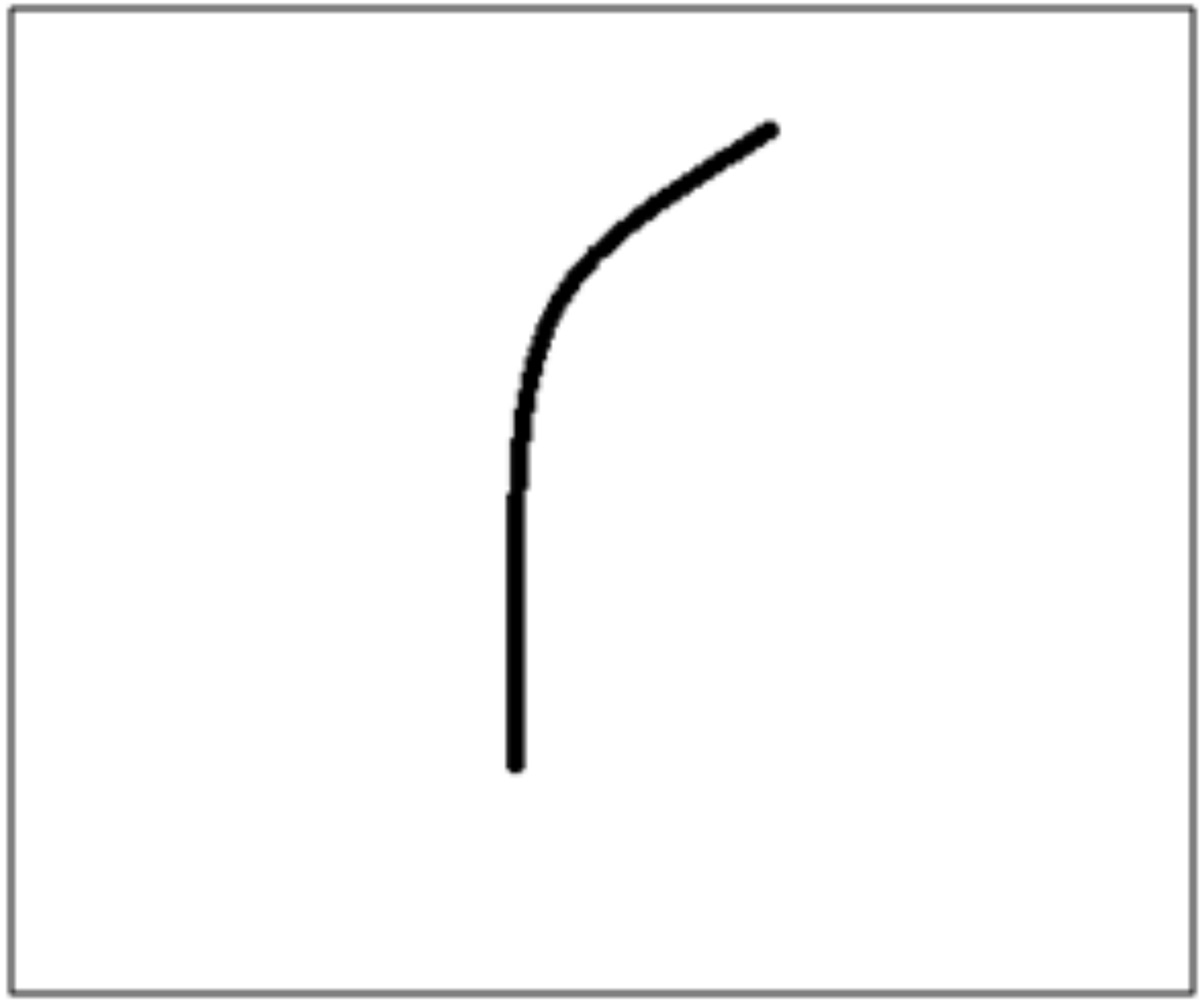


Let’s talk about what this convolution is actually doing from a high level.
We said that each of these filters can be thought of as **feature identifiers.**

Let’s say our filter (7x7) is going to be a **curve detector**. As a curve detector, the filter will have a pixel structure in which there will be higher numerical values along the area that is a shape of a curve.

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

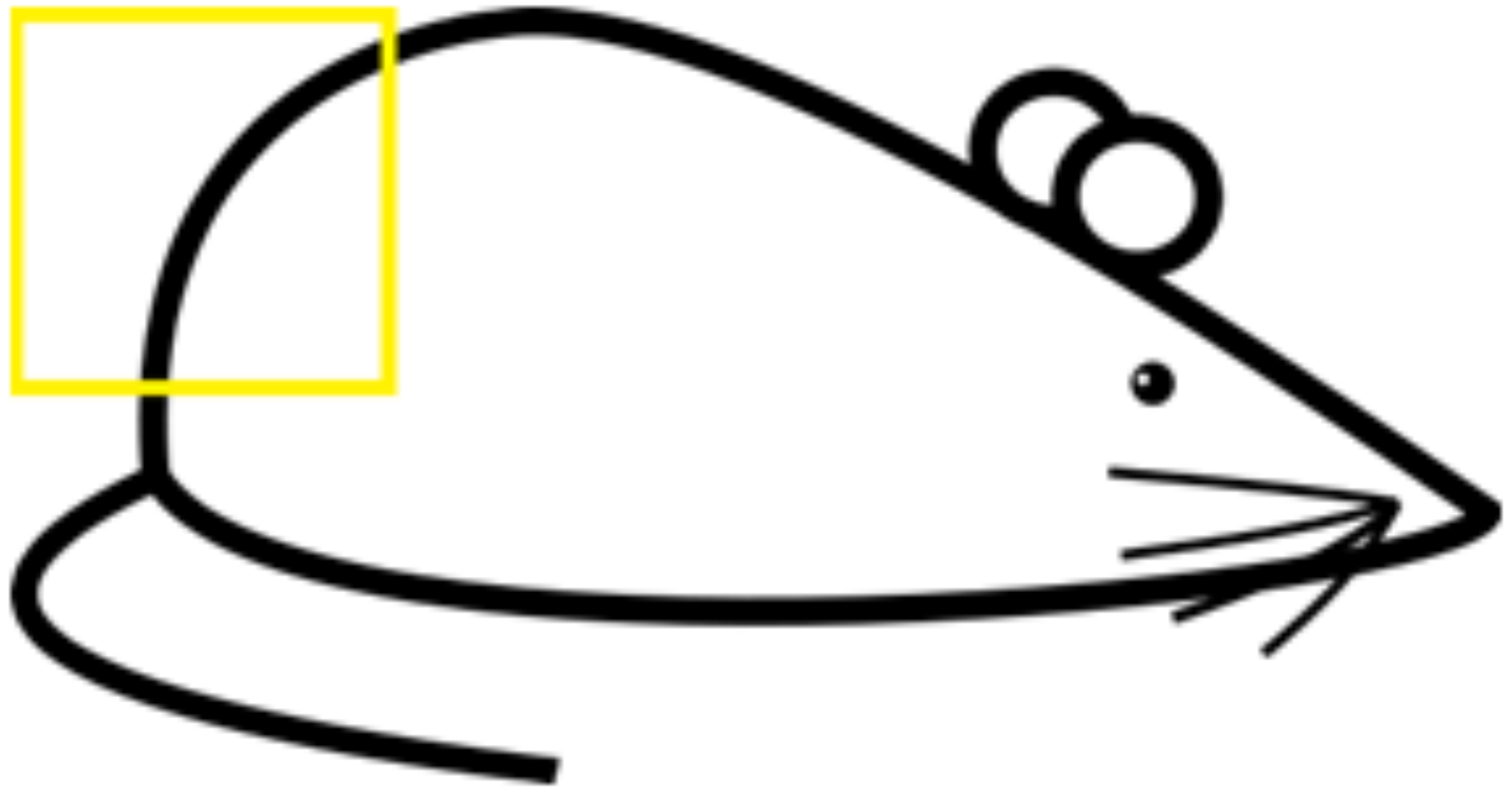
Pixel representation of filter



Visualization of a curve detector filter

Now let's take an example of an image that we want to classify, and let's put our filter at the top left corner.

Original image



Visualization of the filter on the image

We have to do is multiply the values in the filter with the original pixel values of the image.

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Now let’s see what happens when we move our filter…

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

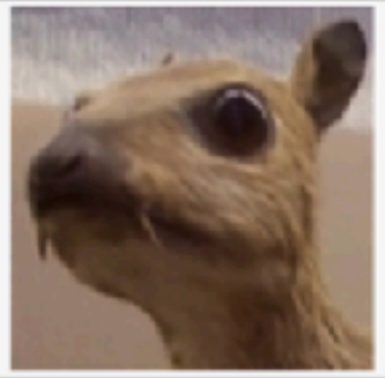
Pixel representation of receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

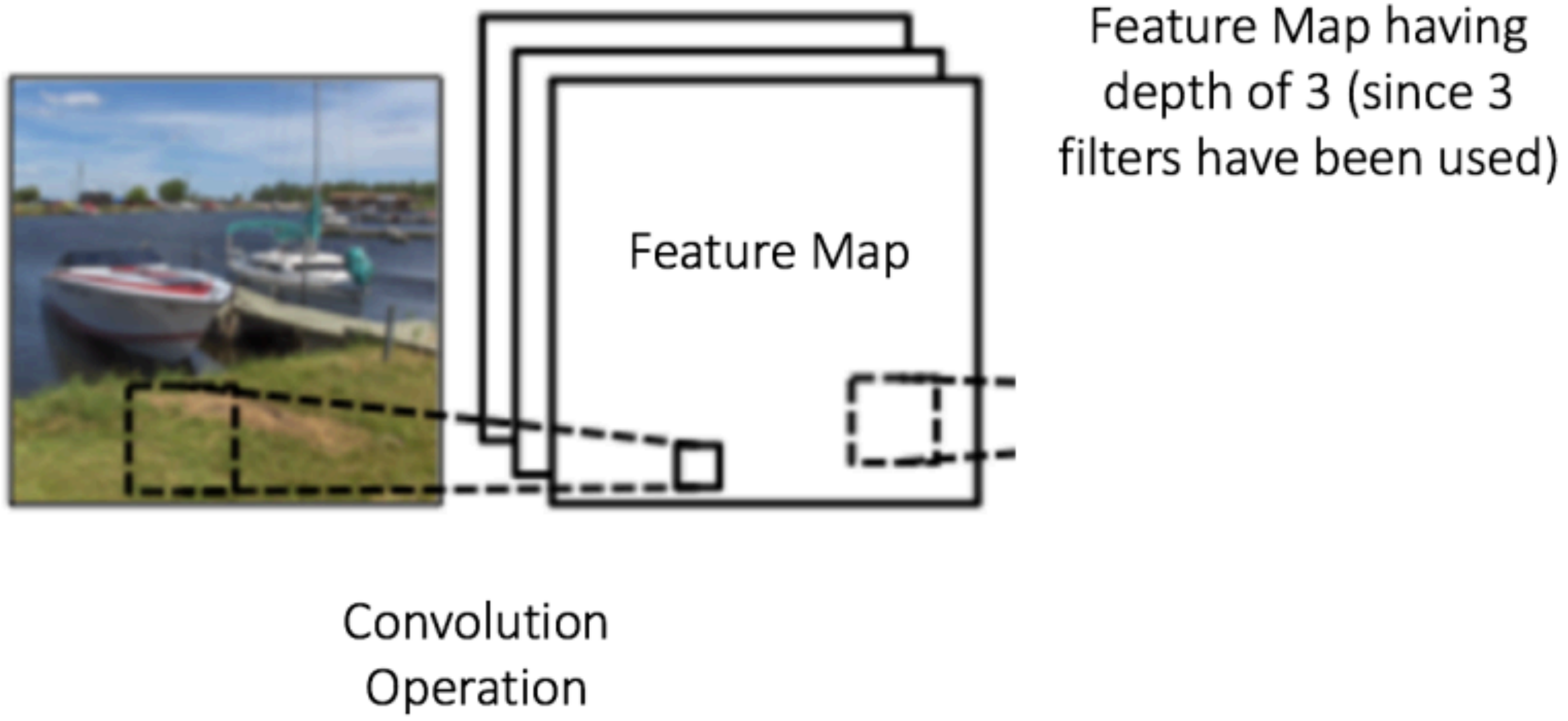
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Different values of the filter matrix will produce different **Feature Maps** for the same input image.

In this example a filter (with red outline) slides over the input image (convolution operation) to produce a feature map.
The convolution of another filter (with the green outline), over the same image gives a different feature map as shown.

In practice, a **CNN *learns* the values of these filters on its own during the training** process (although we still need to specify parameters such as number of filters, filter size, architecture of the network etc. before the training process).

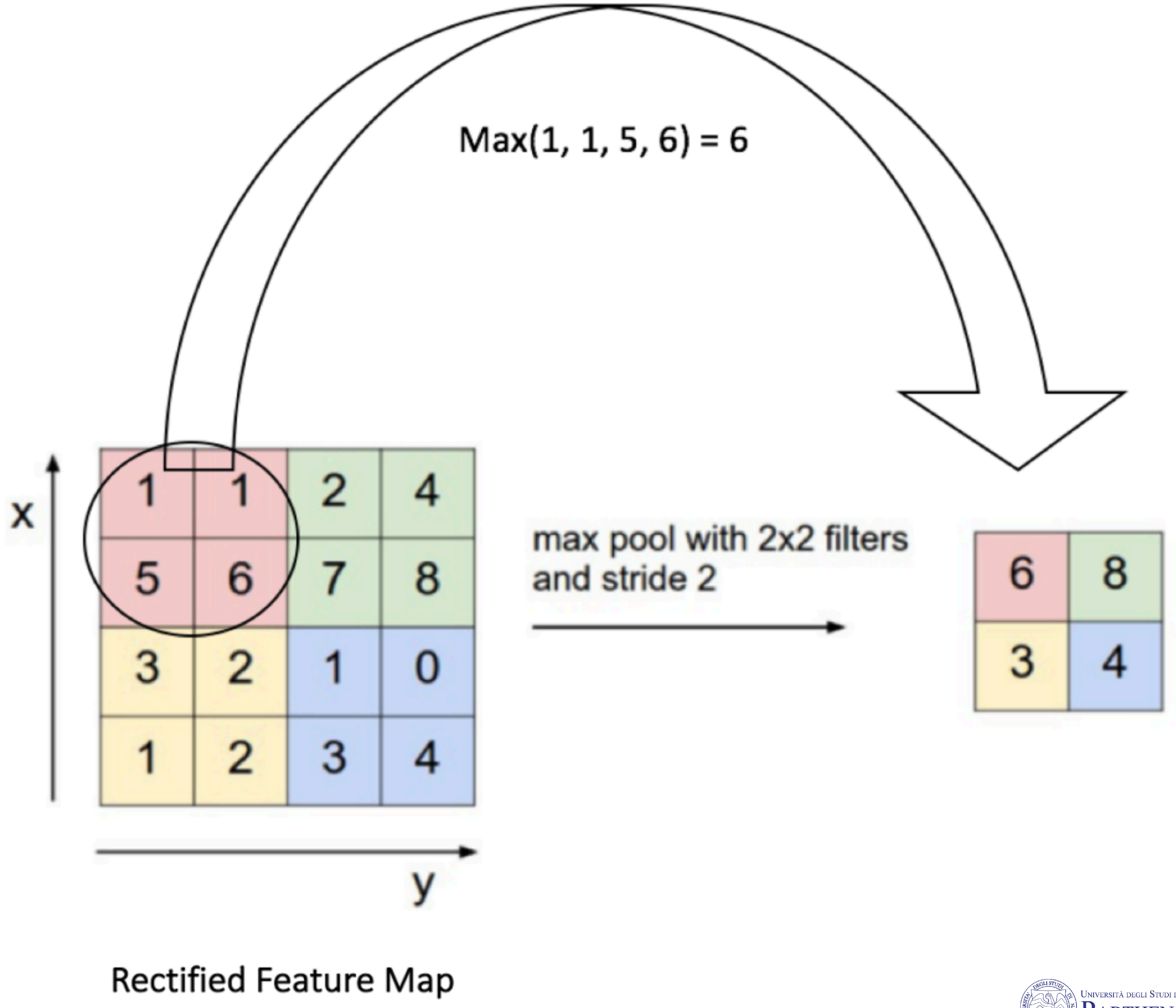
The more number of filters we have, the more image features get extracted and the better our network becomes at recognizing patterns in unseen images.



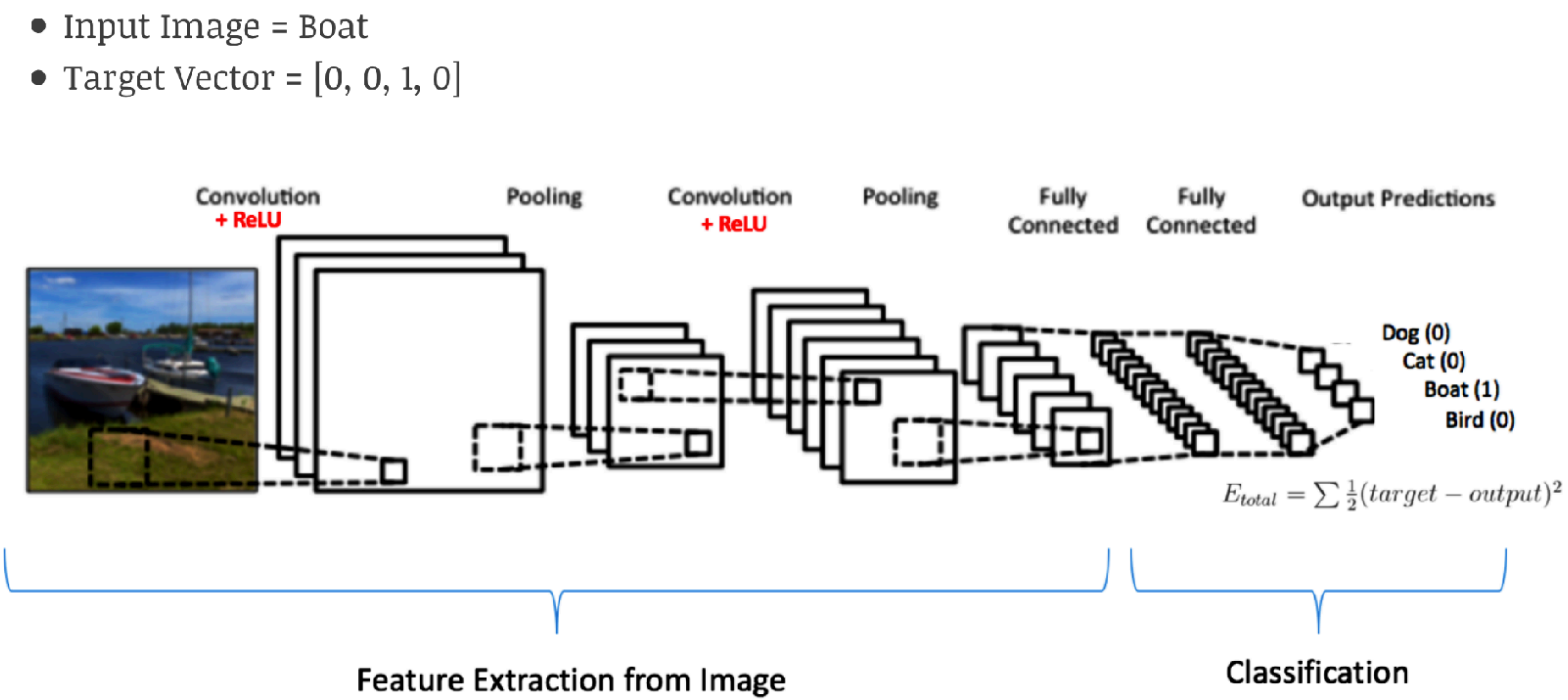
In a traditional CNN architecture,
there are other layers that are in between these conv layers: the **pooling** step

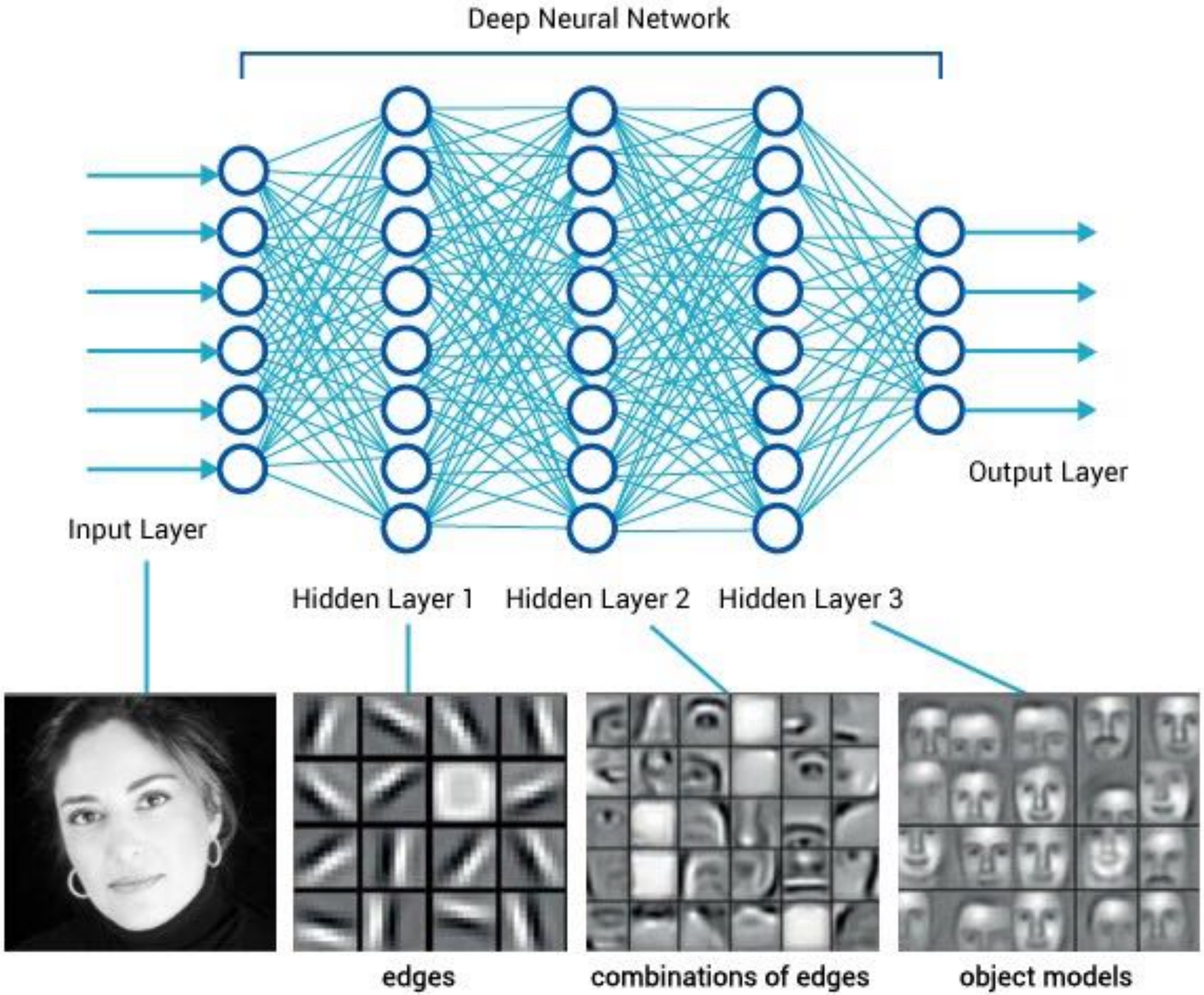
It reduces the dimensionality of each feature map but retains the most important information.

Spatial Pooling can be of different types: Max, Average, Sum etc.

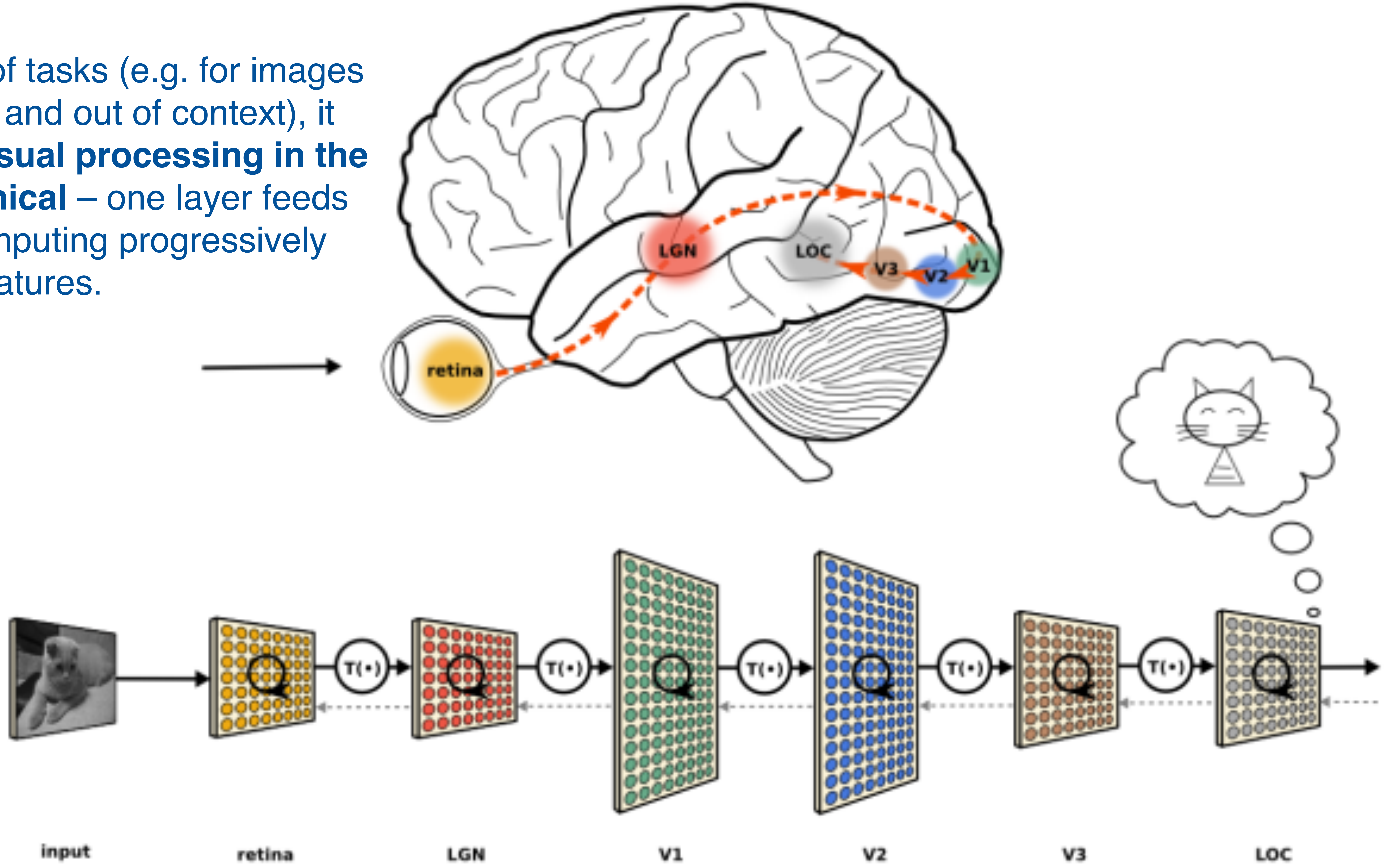


Together these layers extract the useful features from the images, introduce non-linearity in the network and reduce feature dimension while aiming to make the features somewhat equivariant to scale and translation

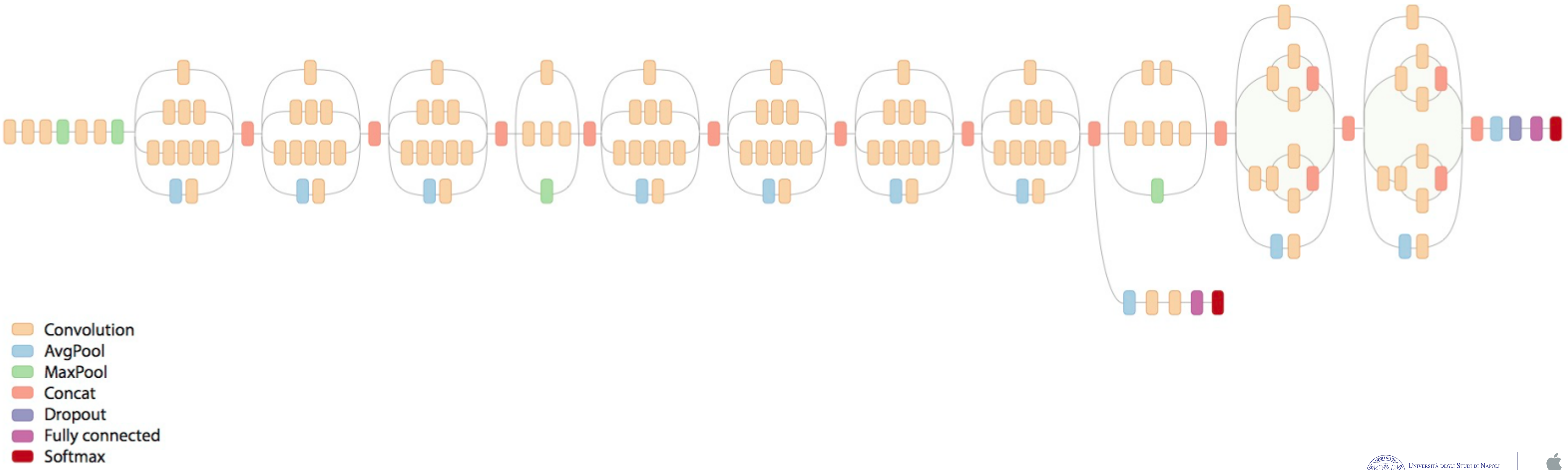



Deep Neural Network
Input Layer
Hidden Layer 1
Hidden Layer 2
Hidden Layer 3
Output Layer
edges
combinations of edges
object models

For some types of tasks (e.g. for images presented briefly and out of context), it is thought that **visual processing in the brain is hierarchical** – one layer feeds into the next, computing progressively more complex features.



Inception V3 network (model)



Tabby, tabby cat

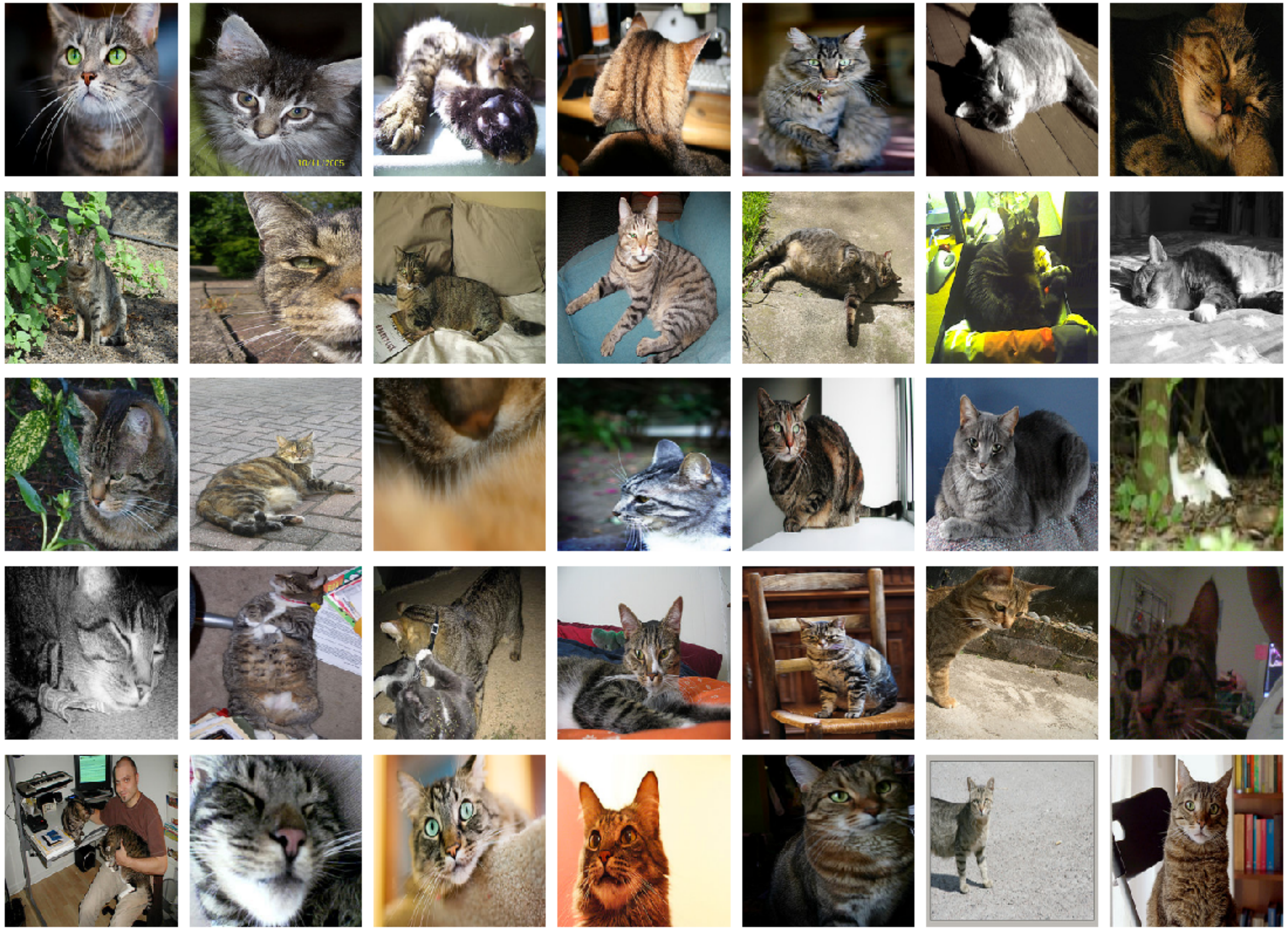
A cat with a grey or tawny coat mottled with black

1525 pictures | 58.3% Popularity Percentile | Wordnet IDs

Numbers in brackets: (the number of synsets in the subtree).

- ImageNet 2011 Fall Release (32326)
  - plant, flora, plant life (4486)
  - geological formation, formation (17
  - natural object (1112)
  - sport, athletics (176)
  - artifact, artefact (10504)
  - fungus (308)
  - person, individual, someone, someb
  - animal, animate being, beast, brute
    - invertebrate (766)
    - homeotherm, homoiotherm, hor
    - work animal (4)
    - darter (0)
    - survivor (0)
    - range animal (0)
    - creepy-crawly (0)
    - domestic animal, domesticated
      - domestic cat, house cat, Feli
        - Egyptian cat (0)
        - Persian cat (0)
        - kitty, kitty-cat, puss, puss
        - tiger cat (0)
        - Angora, Angora cat (0)
        - tom, tomcat (1)
        - Siamese cat, Siamese (1)
        - Manx, Manx cat (0)
        - Maltese, Maltese cat (0)
        - tabby, queen (0)
        - Burmese cat (0)
        - alley cat (0)

Treemap Visualization | Images of the Synset | Downloads



*Images of children synsets are not included. All images shown are thumbnails. Images may be subject to copyright.

Prev 1 2 3 4 5 6 7 8 9 10 ... 67 68 Next

Computer Vision demo app

and
now
testing
time…

and now ?...

Università degli Studi di Napoli
Parthenope

Foundation
Program

AI on smart phones


Killer apps, not killer robots will define AI's contribution to the world.
Andrew Ng. Computer scientist founder of deeplearning.ai

available sensors on my device

Which data can I process in my app?

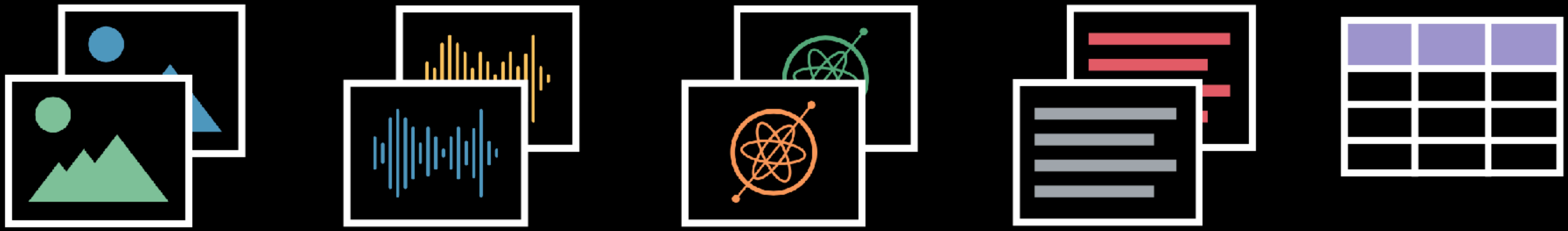
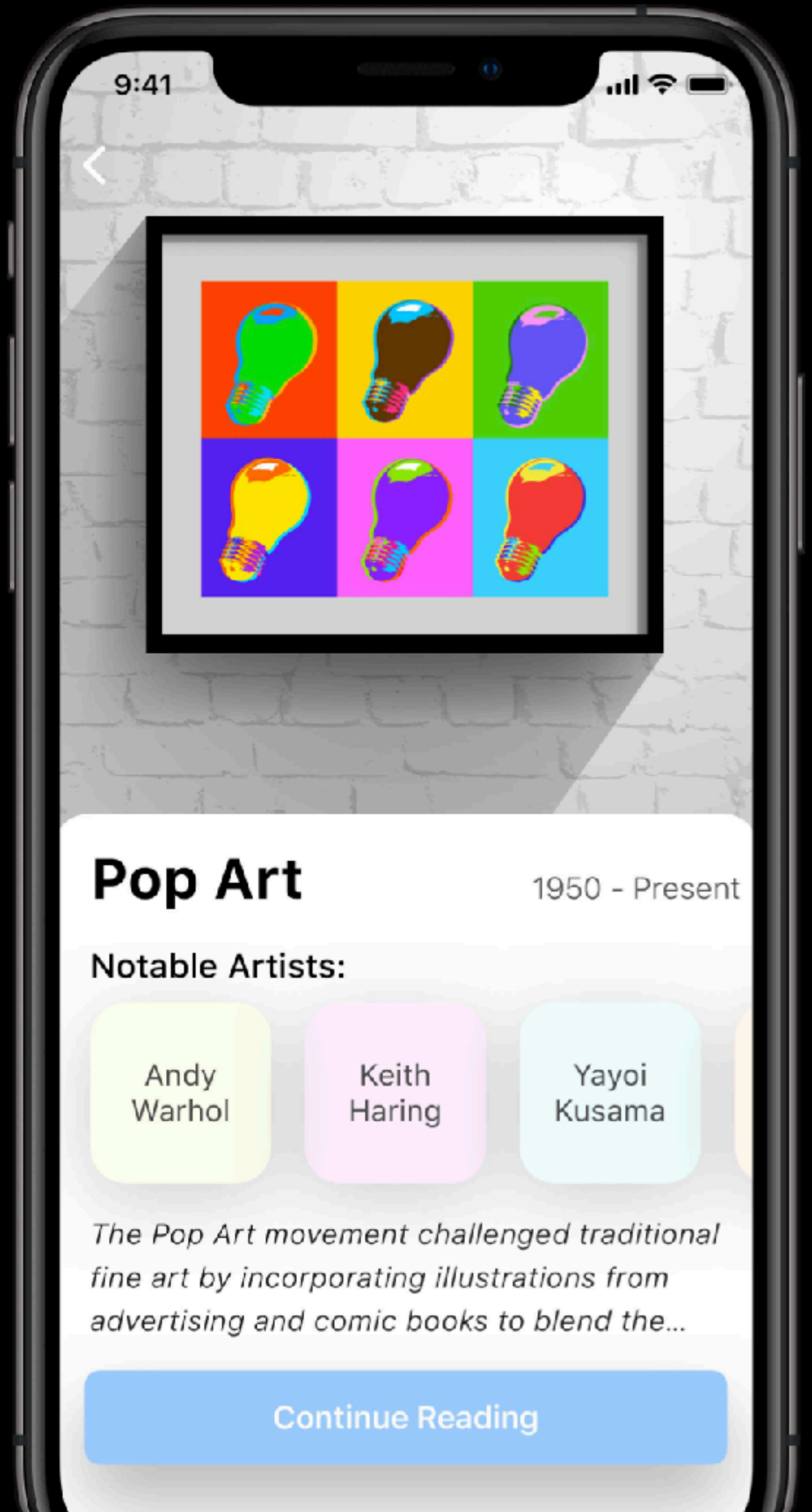


Image | Sound | Activity | Text | Tabular

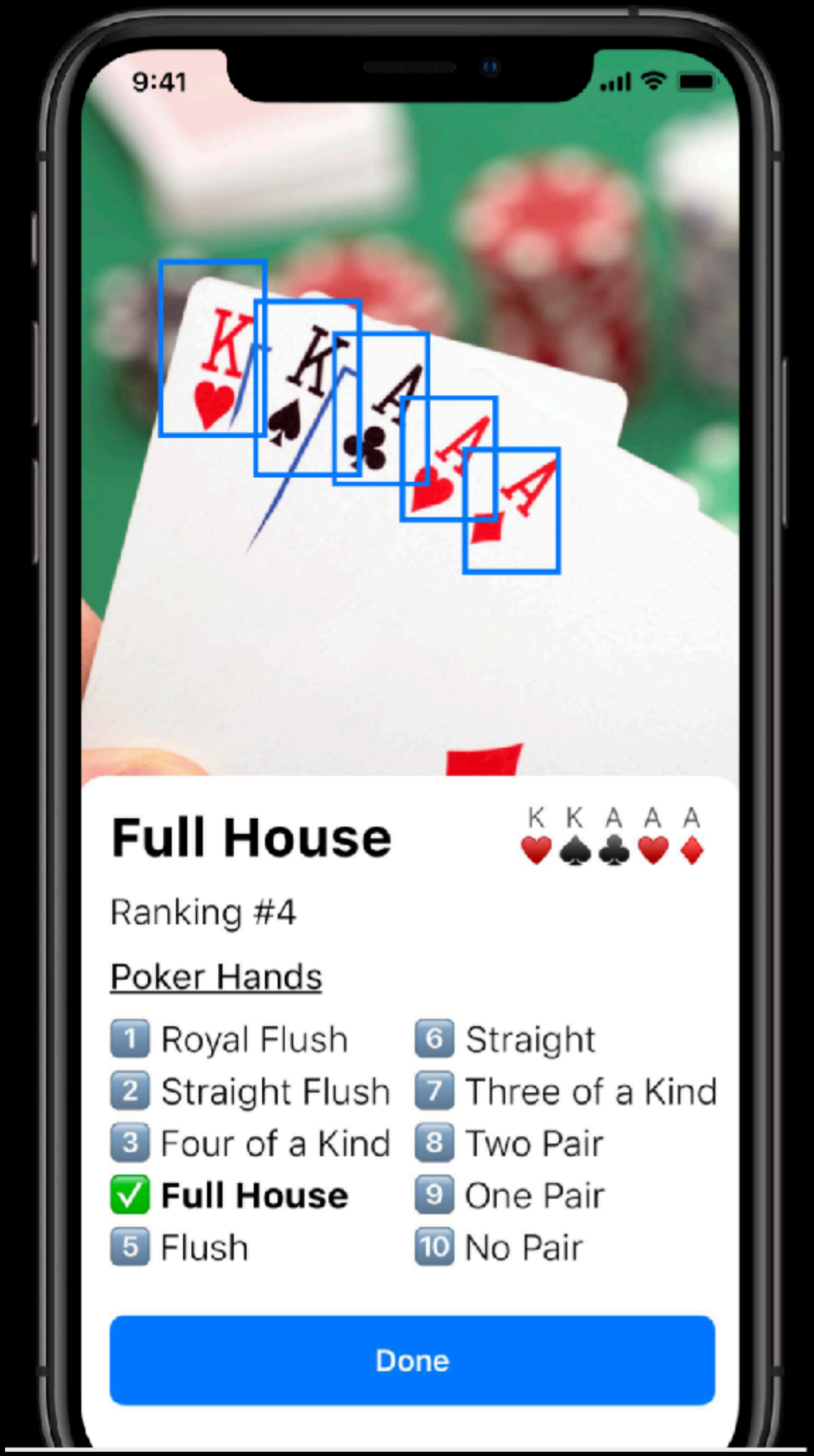
Image Classification

Classify an image based on its content



Object Detection

Localize and recognizes content in an image



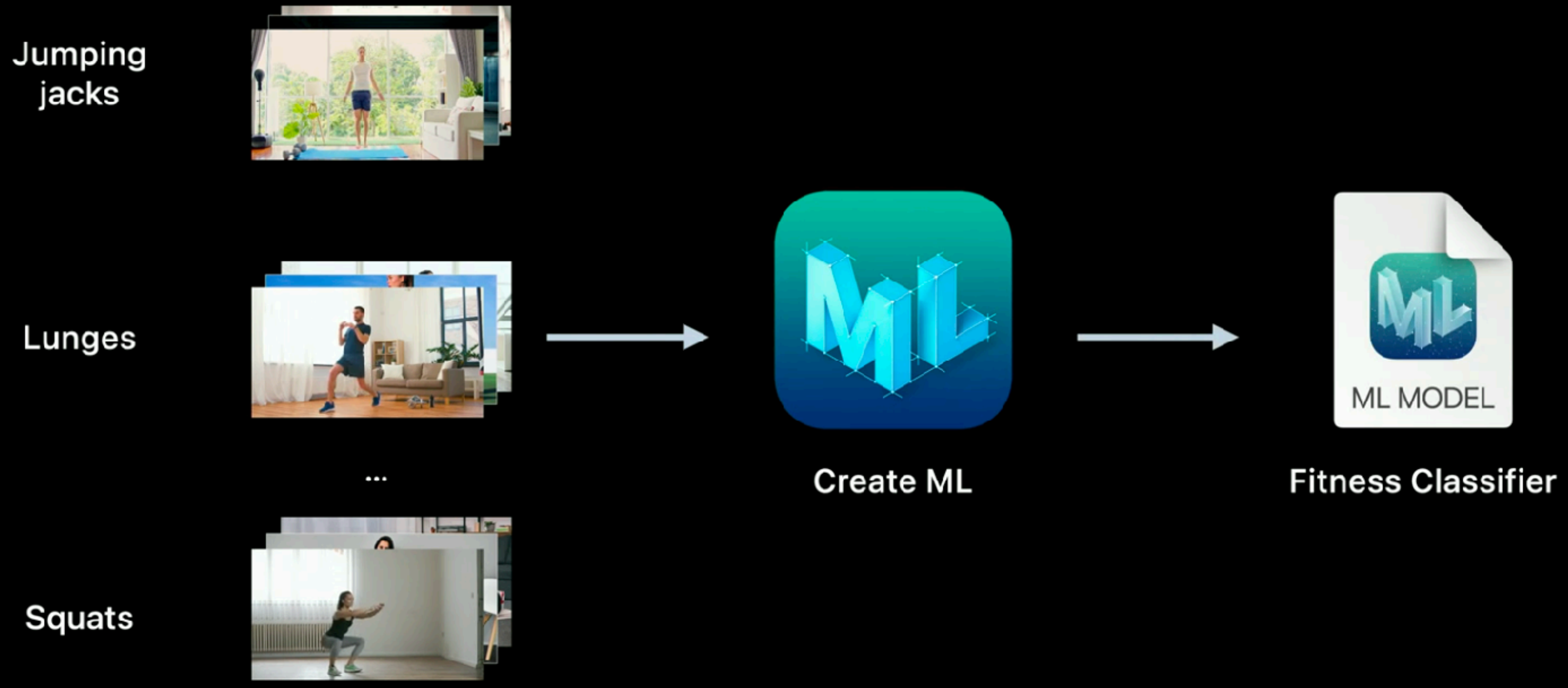
Sound classification

Categorize contents of audio

Activity Classifier

Categorize contents
of motion (data from sensors)

Activity Classifier

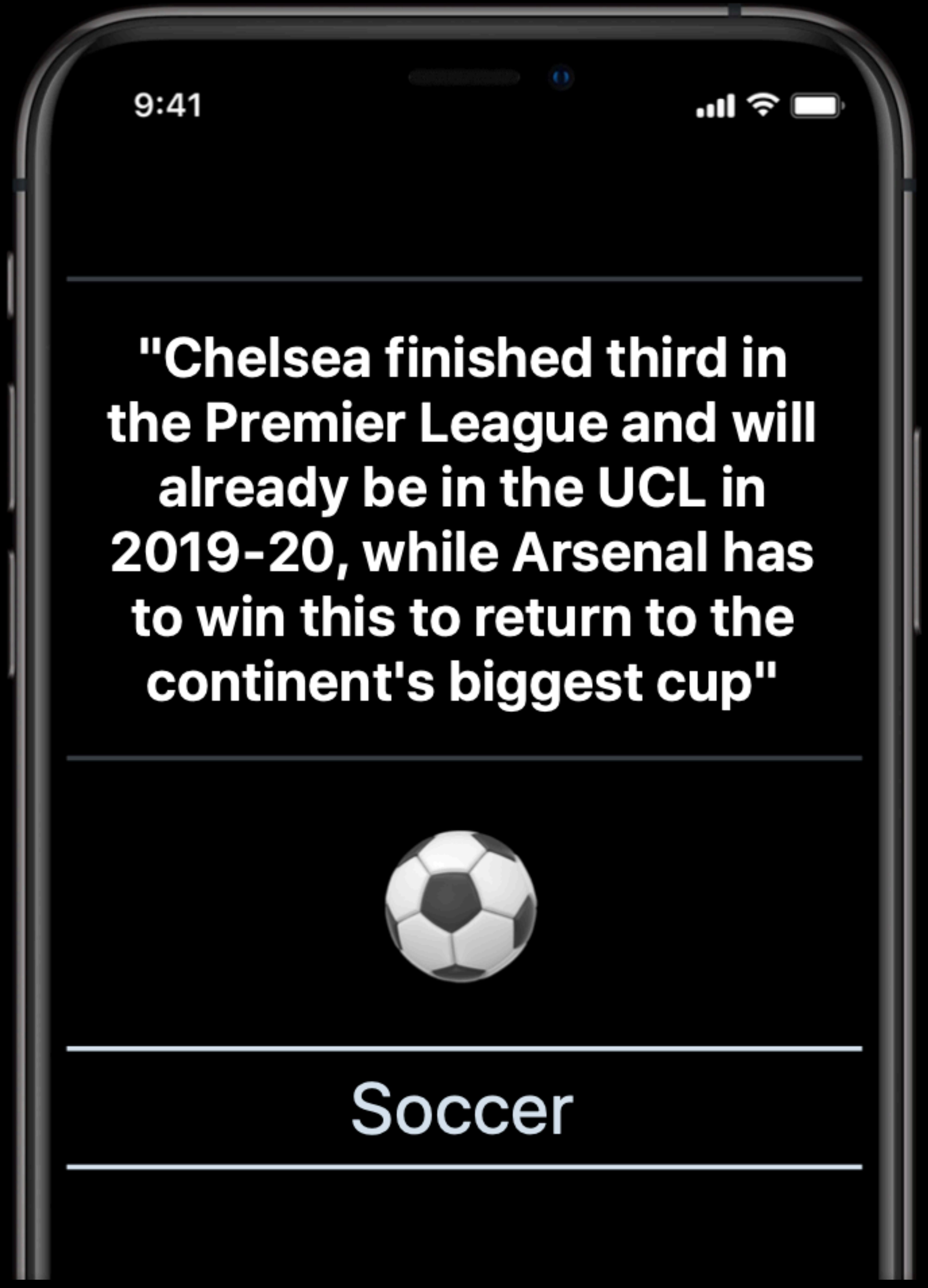


Classify actions directly from video data

Only available on Mac OS 11 Big Sur

Text Classifier

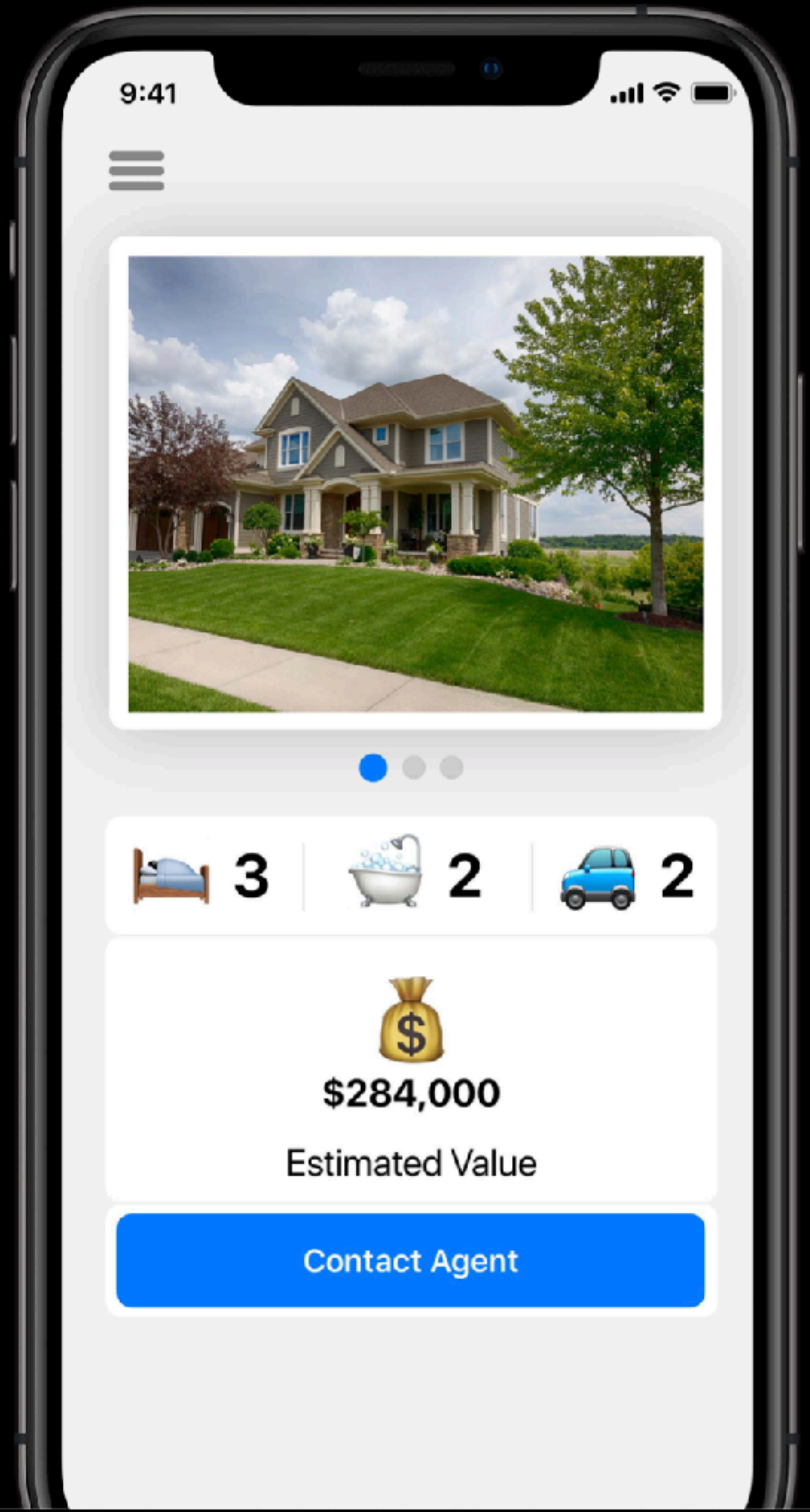
Labels text based
on its content



Sentiment Analysis

Tabular Regressor

Predict a value
by features of interest



Recommender

Reccommends content based on behavior

Style transfer

Style

+

Content

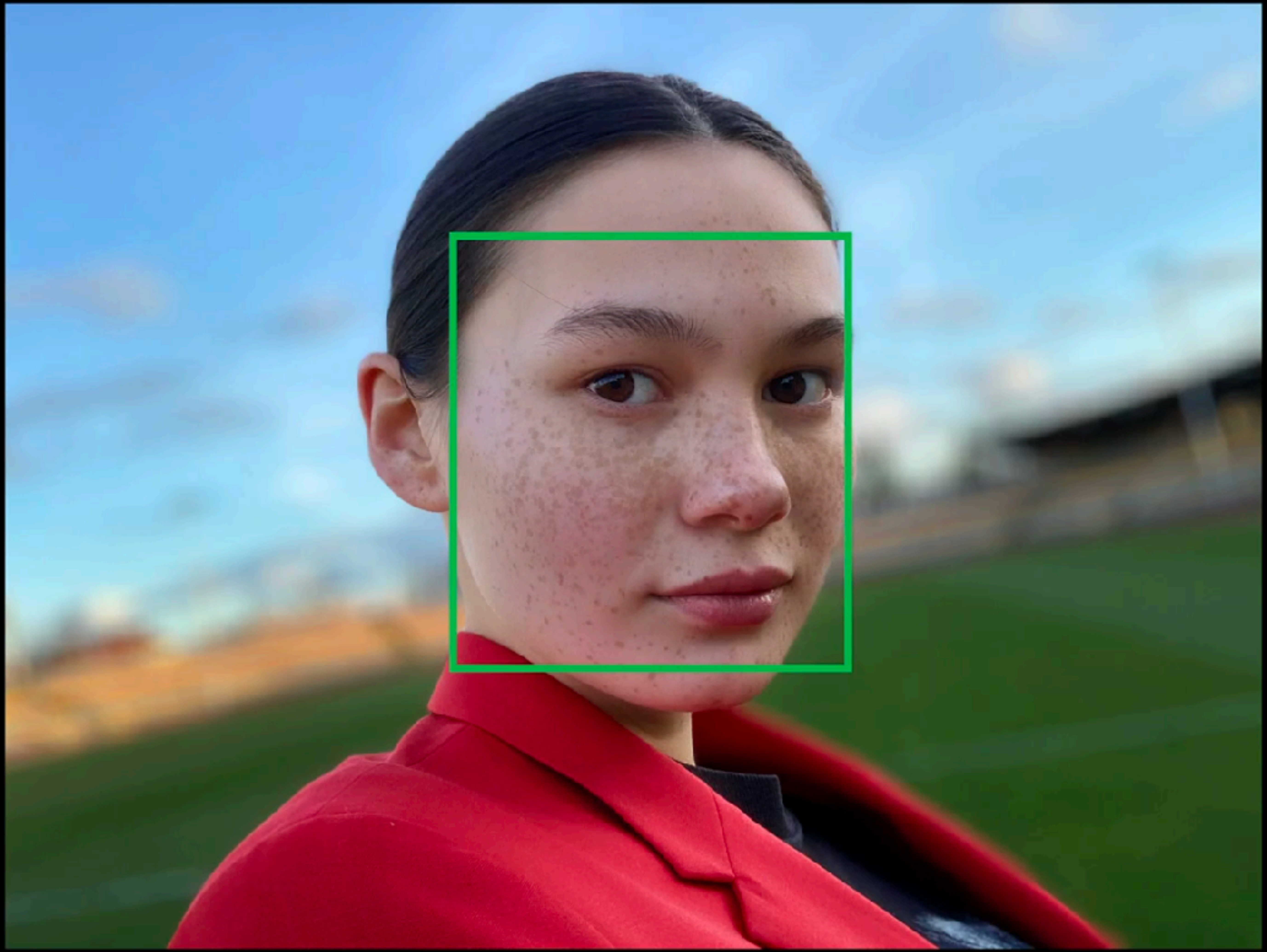
Only available on Mac OS 11 Big Sur

Style transfer

Stylized result

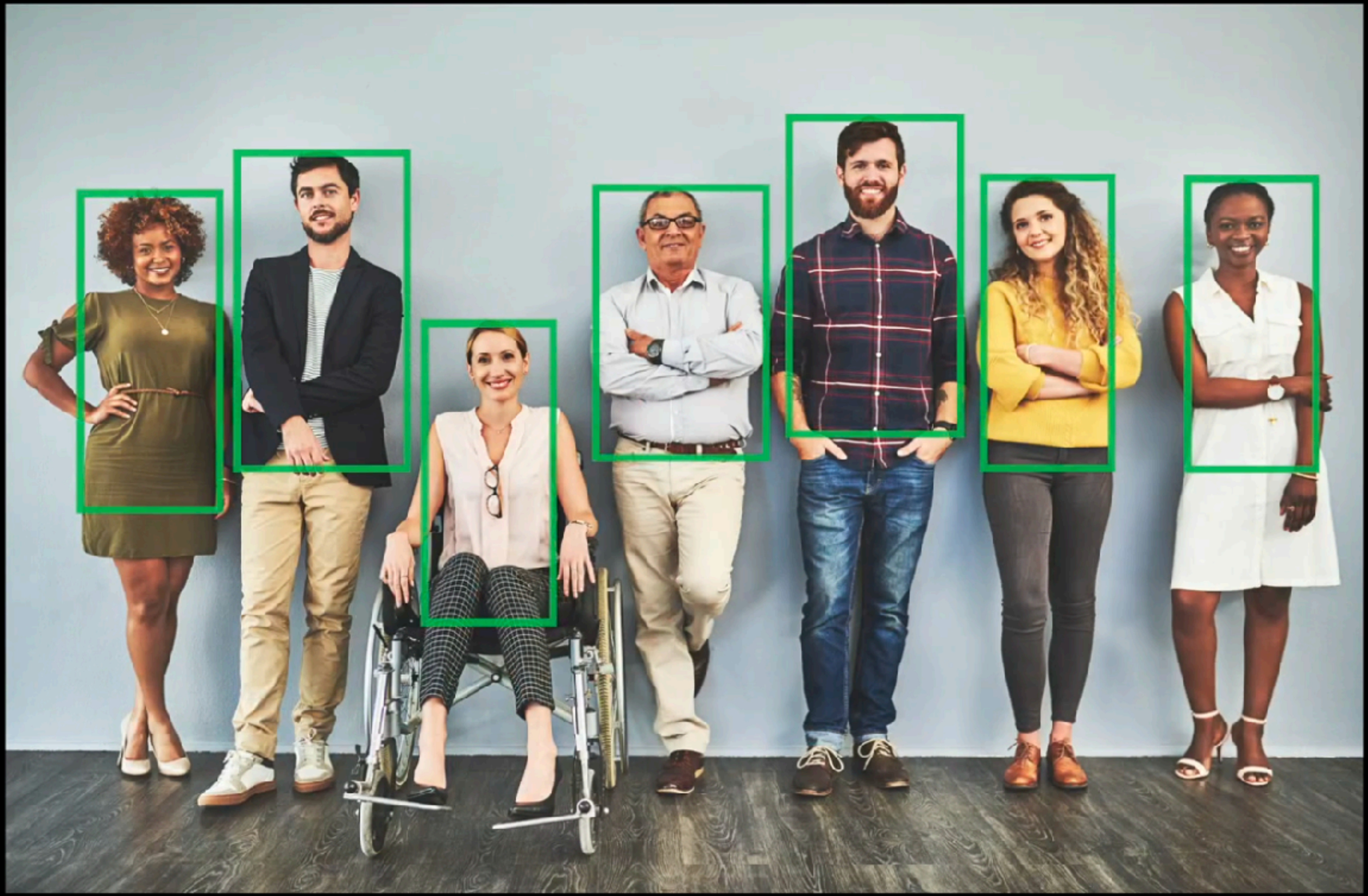
Other computer vision tasks . . .

Face Detection



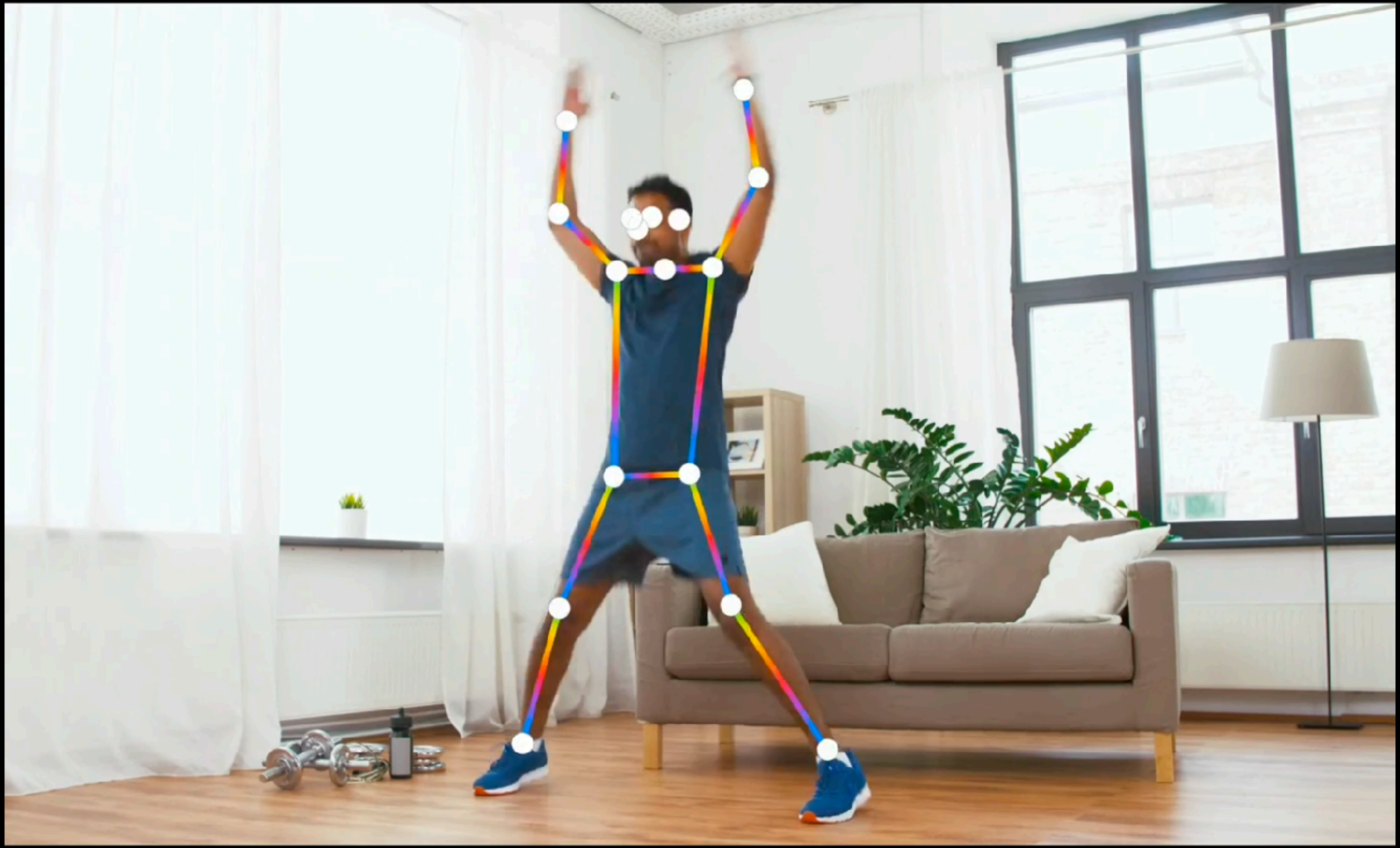
Face Landmarks

Human Detector

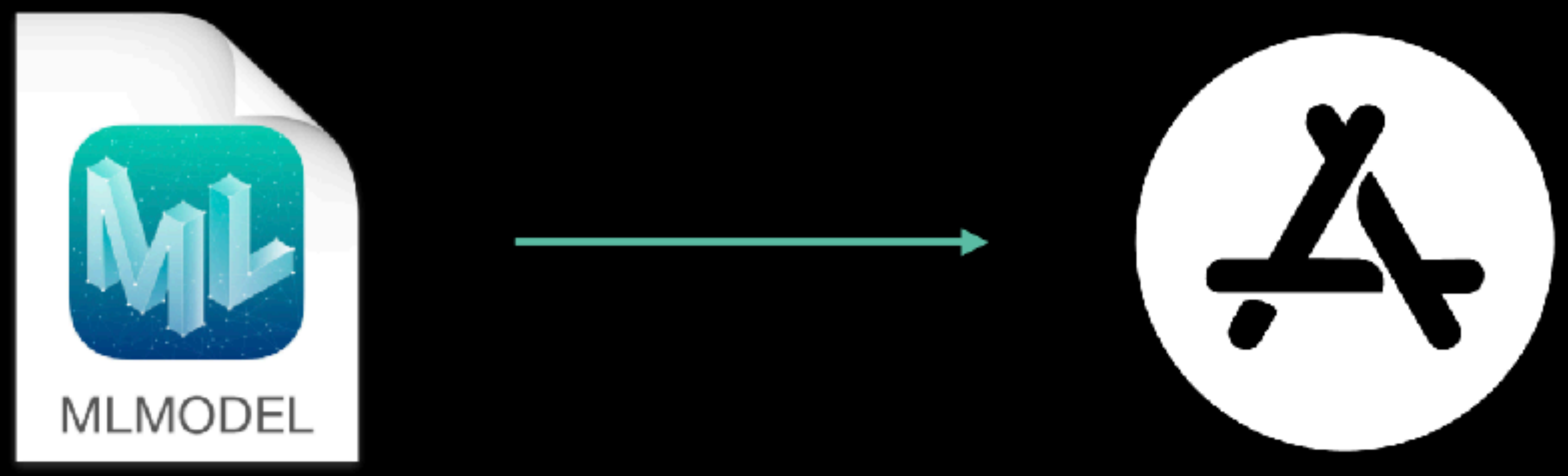


Hand pose estimation

Body pose estimation



Core ML Framework

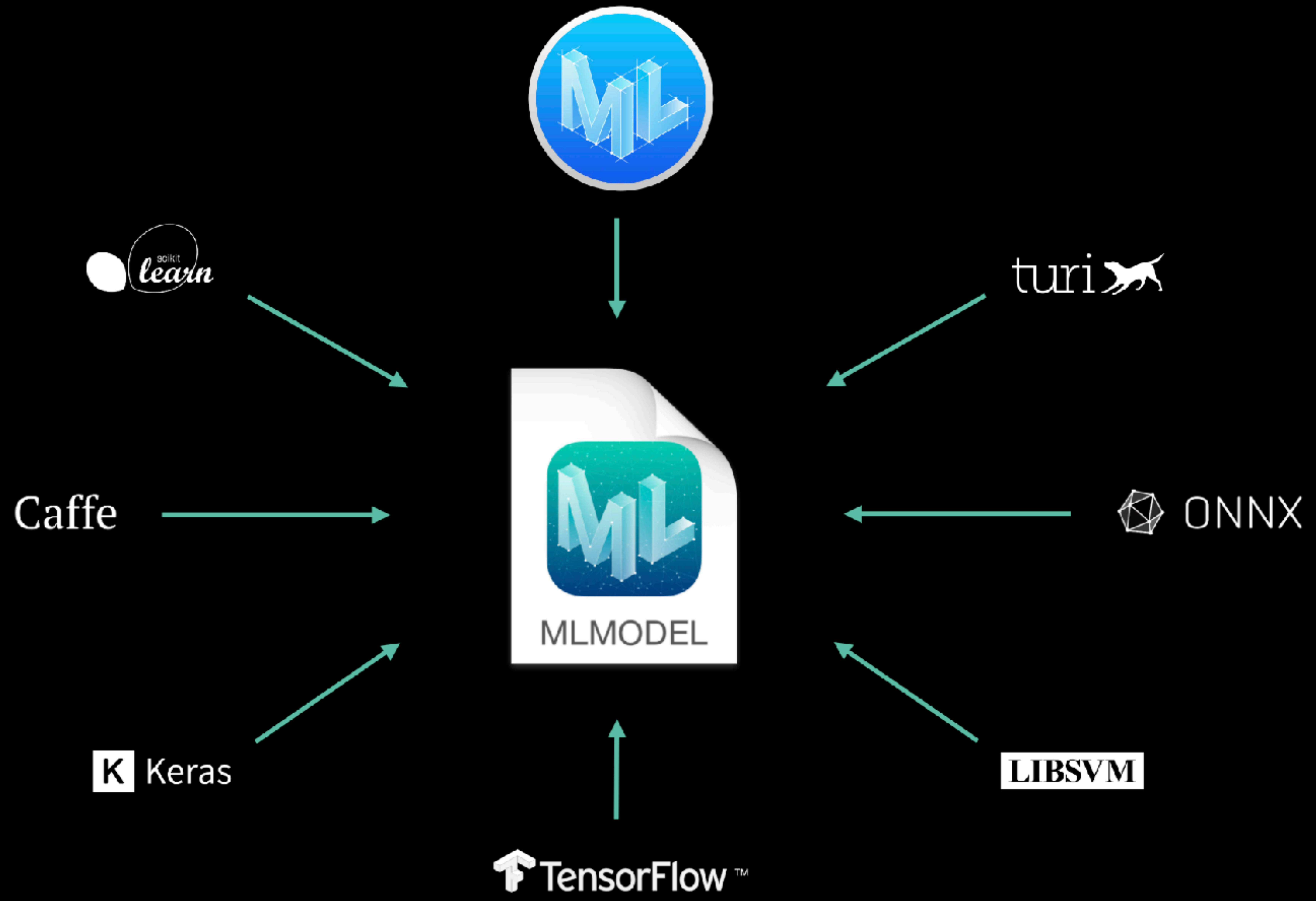

MLMODEL

On-Device

Privacy

No server

Available


scikit learn
turi
Caffe
ONNX
ML
MLMODEL
K Keras
LIBSVM
TensorFlow™

Available Algorithms

Sentiment Analysis

Handwriting Recognition

Translation

Scene Classification

Style Transfer

Music Tagging

Predicting Text

Feed Forward Neural Networks	Convolutional Neural Networks	Recurrent Neural Networks
Tree Ensembles	Support Vector Machines	Generalized Linear Models

Focus on Use Case

Sentiment Analysis

Handwriting Recognition

Translation

Scene Classification

Style Transfer

Music Tagging

Predicting Text

Sample Models

https://developer.apple.com/machine-learning

Core ML models

Ready to use

Task specific

Explore!

Places205-GoogLeNet

Detects the scene of an image from 205 categories such as an airport terminal, bedroom, forest, coast, and more.

View original model details ›

Download Core ML Model

File size: 24.8 MB

ResNet50

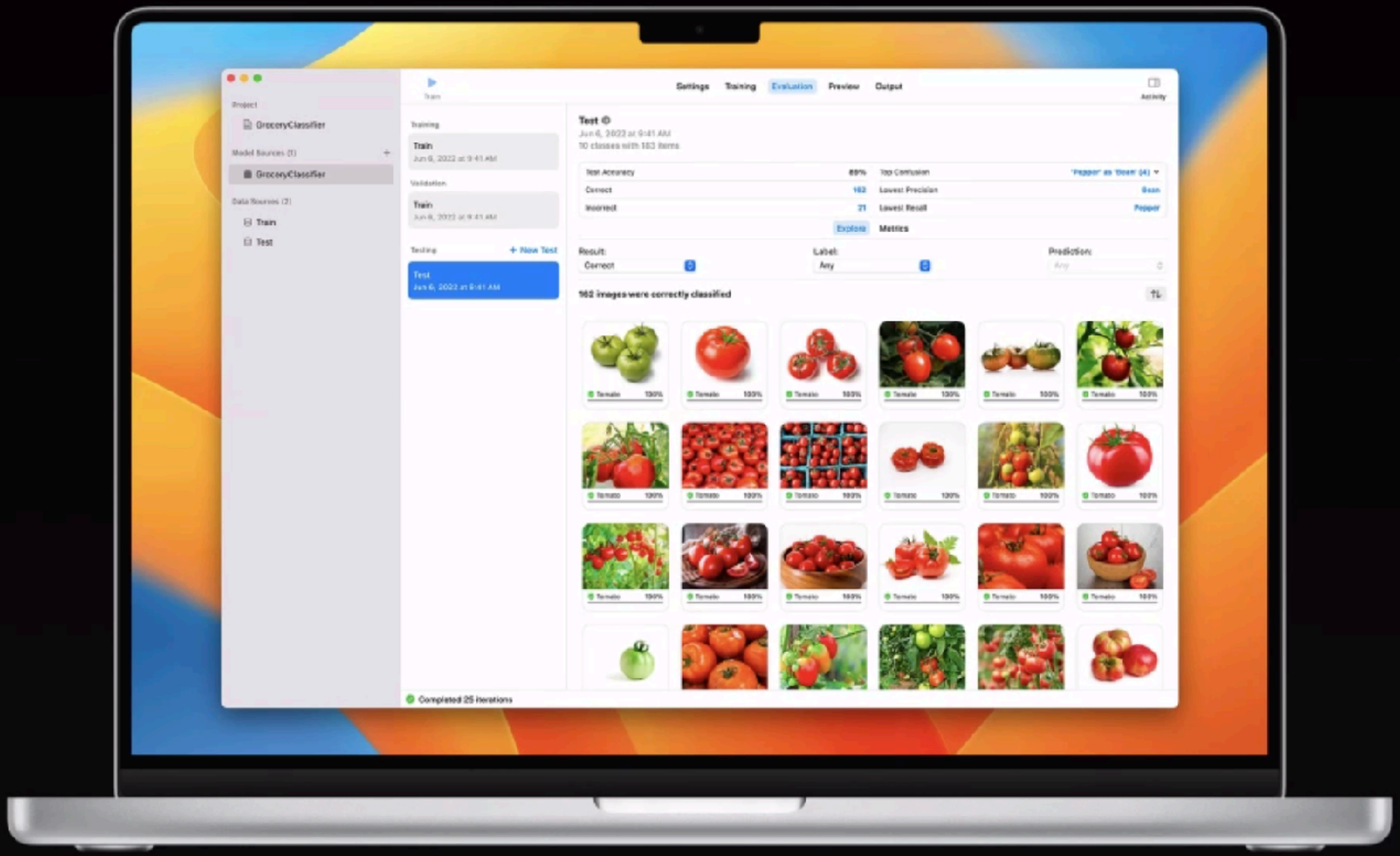
Detects the dominant objects present in an image from a set of 1000 categories such as trees, animals, food, vehicles, people, and more.

View original model details ›

Download Core ML Model

File size: 102.6 MB

CreateML



Create ML Xcode Mac App
2019

Pre-trained models that use Apple's device sensors

Create ML Xcode Mac App
2019

Image

Image classification
Object detection
Hand pose classification
Style transfer

Video

Action classification
Hand action classification
Style transfer

Motion

Activity classification

Sound

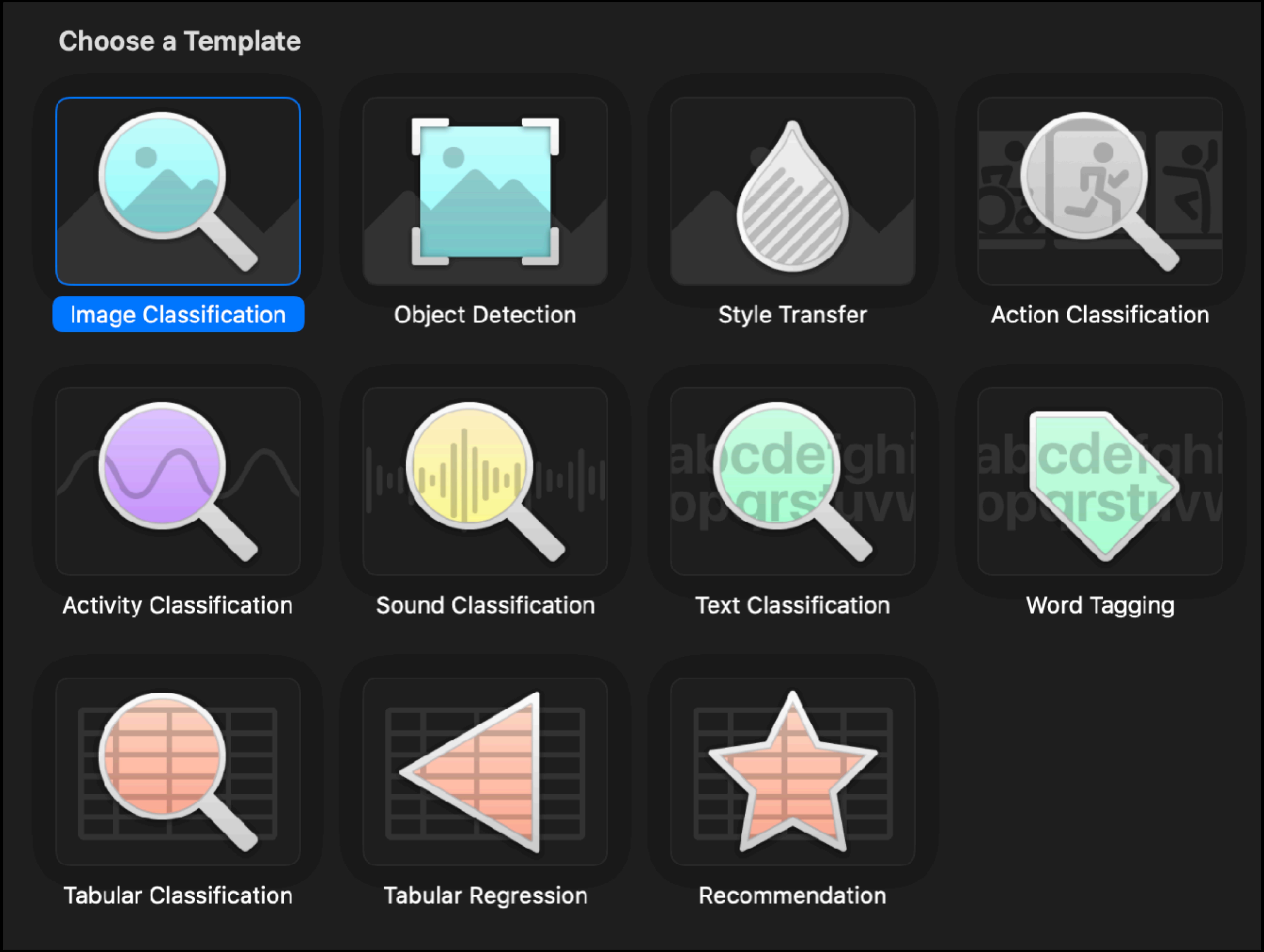
Sound classification

Text

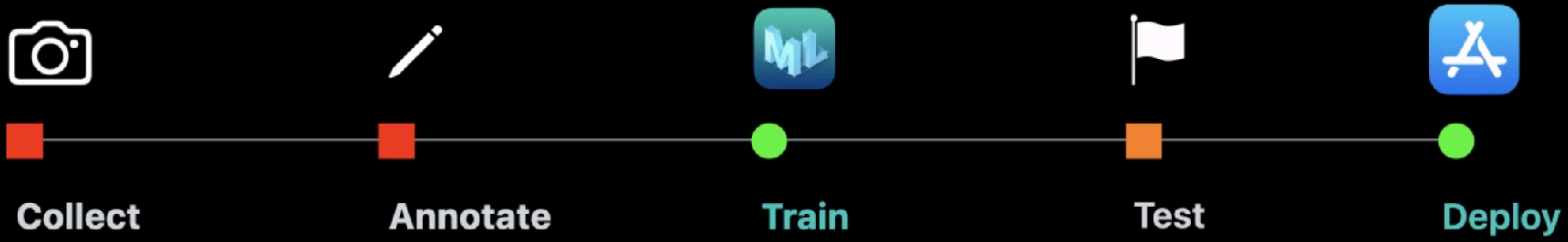
Text classification
Word tagging

Tabular

Tabular classification
Tabular regression
Recommendation

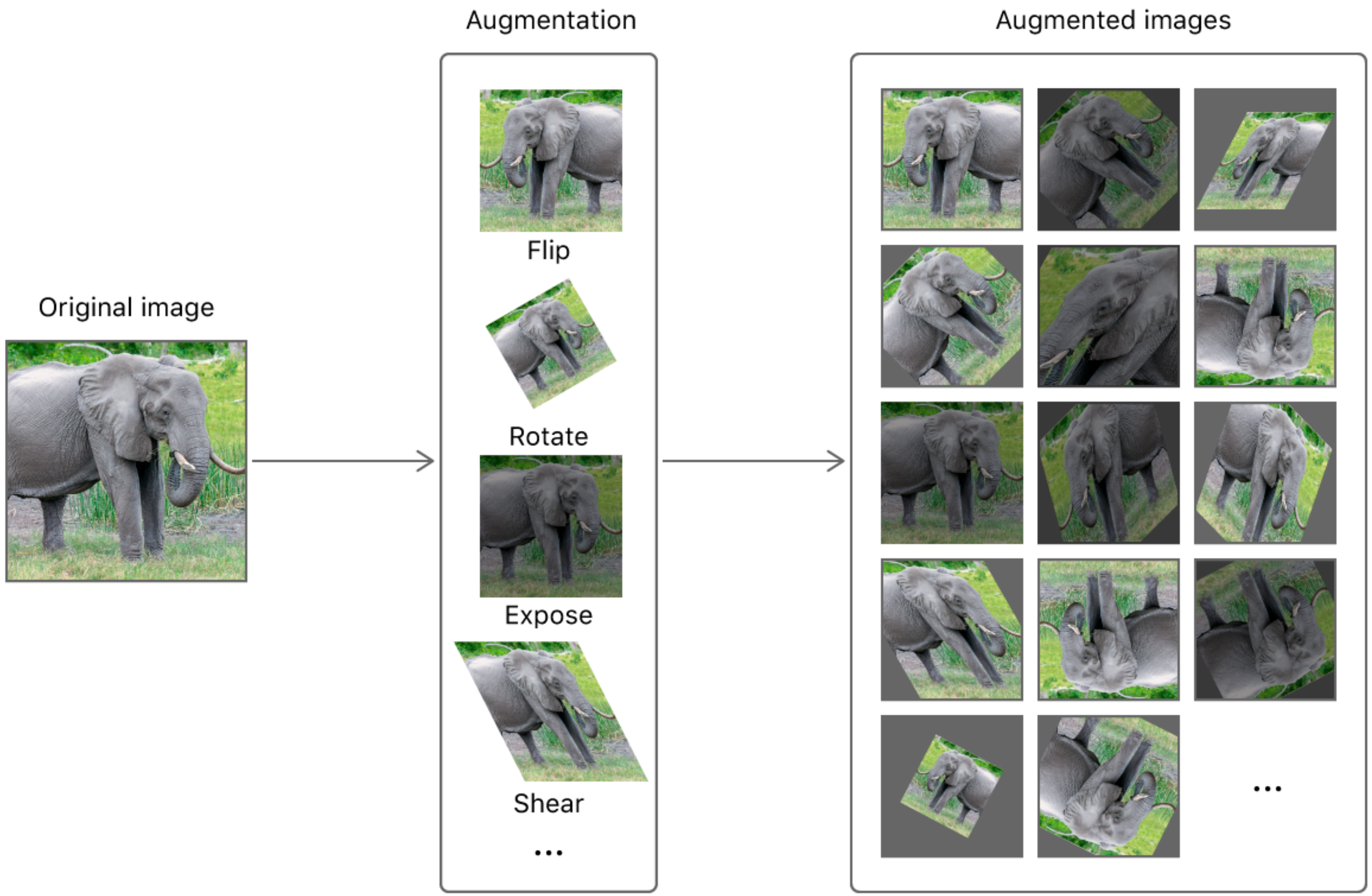

Choose a Template
Image Classification
Object Detection
Style Transfer
Action Classification
Activity Classification
Sound Classification
Text Classification
Word Tagging
Tabular Classification
Tabular Regression
Recommendation

ML model development process



Improving validation accuracy

Increase the amount of data: for image classifiers, you can augment your image data by **flipping, rotating, shearing or changing the exposure** of images.



Make sure the diversity of characteristics of your **training data match those of your testing data,** and both sets are similar to the data your app users will feed to your model.

https://developer.apple.com/documentation/create_ml/improving_your_model_s_accuracy

CHALLENGE

- **Solve a binary classification problem**, starting from a good dataset (search the web for it).
- Organize images (in folders) e clean your data.
- Use CreateML to train the model (image classifier)
- Present your idea
- Timing….(30 mins)

Thanks.