Visual Computing

2024/2025

Class 11

Introduction to Computer Vision

deti departamento de electrónica, telecomunicações e informática

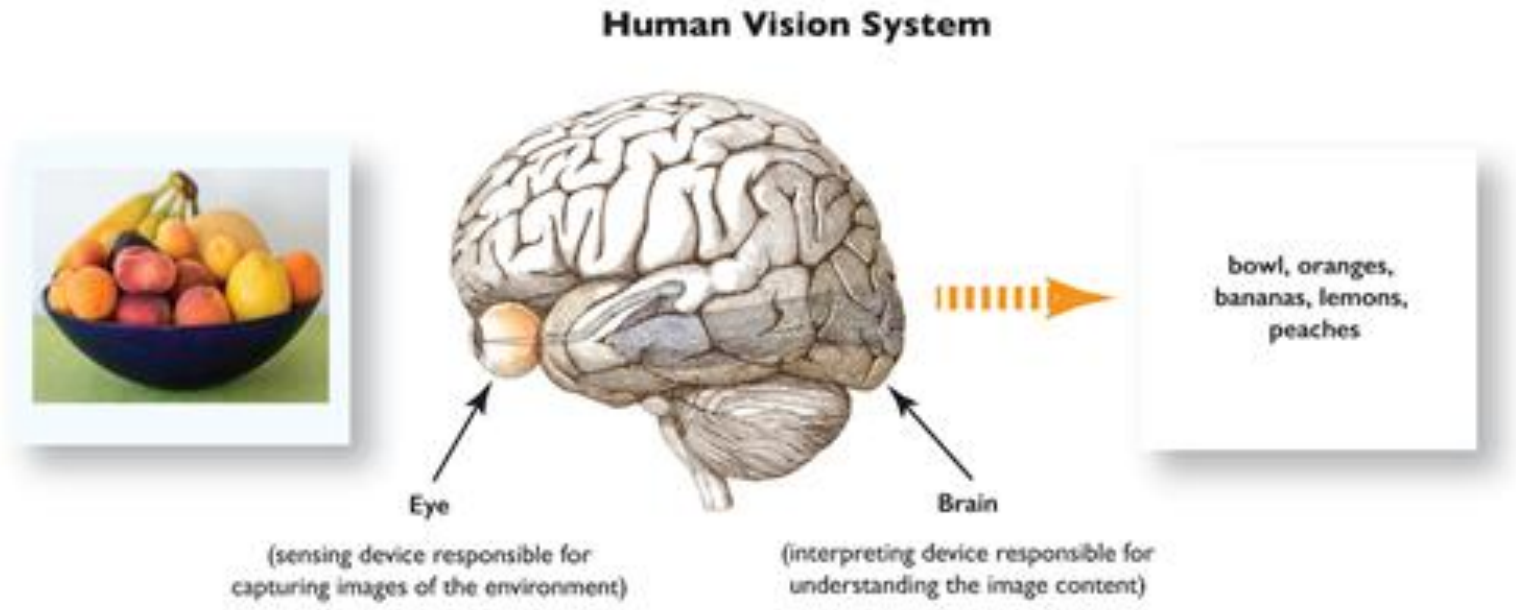
universidade de aveiro

Agenda

- What is Computer Vision
- Image Features
- Basics of object detection
- Approaches to Computer Vision
- Applications
- Off-the-shelf Tools

Human Vision

Making sense of what we see



https://freecontent.manning.com/mental-model-graphic-grokking-deep-learning-for-computer-vision/

Human Vision

Making sense of what we see

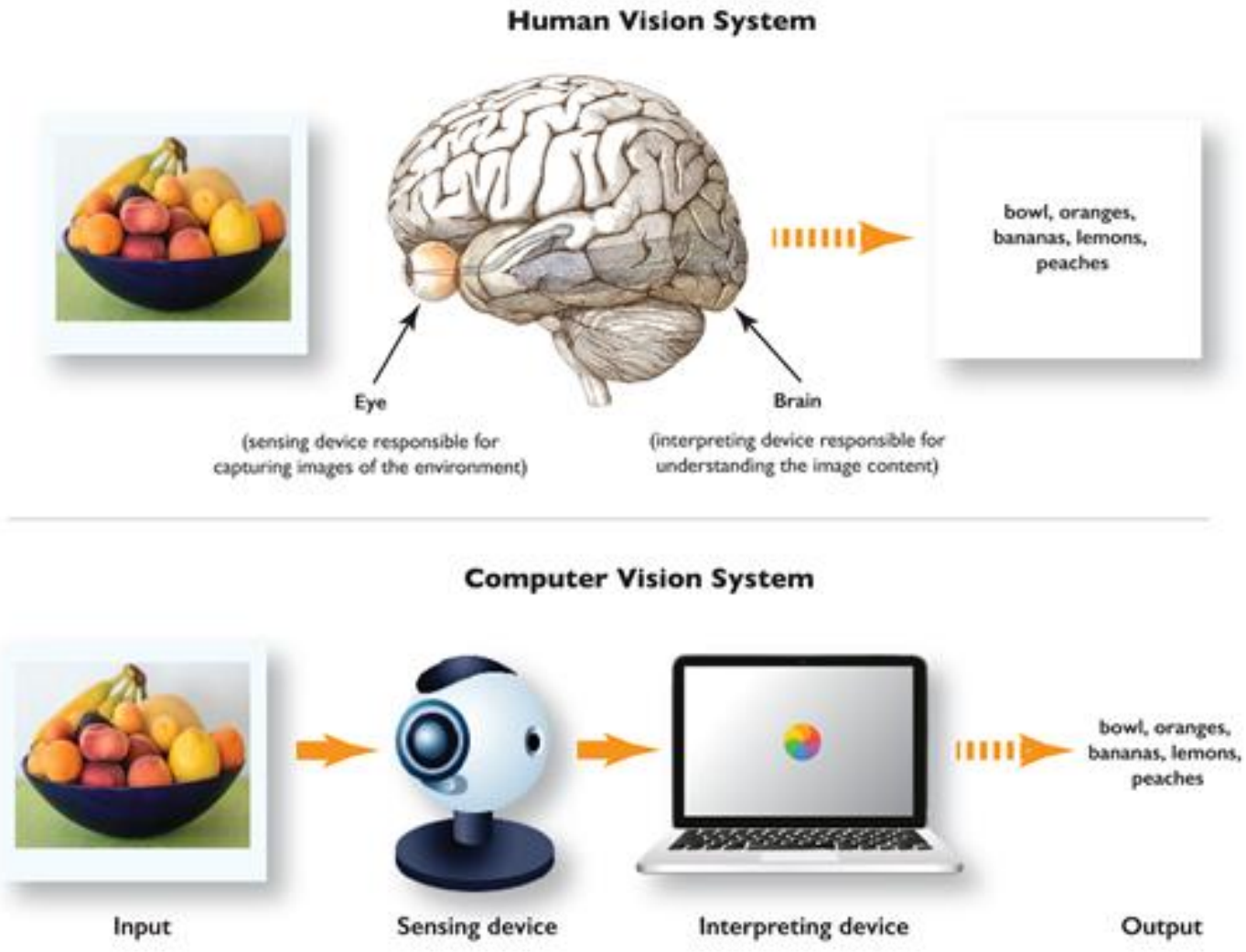
Human Vision

Making sense of what we see

Human Vision

Making sense of what we see

Rise of Computer Vision



https://freecontent.manning.com/mental-model-graphic-grokking-deep-learning-for-computer-vision/

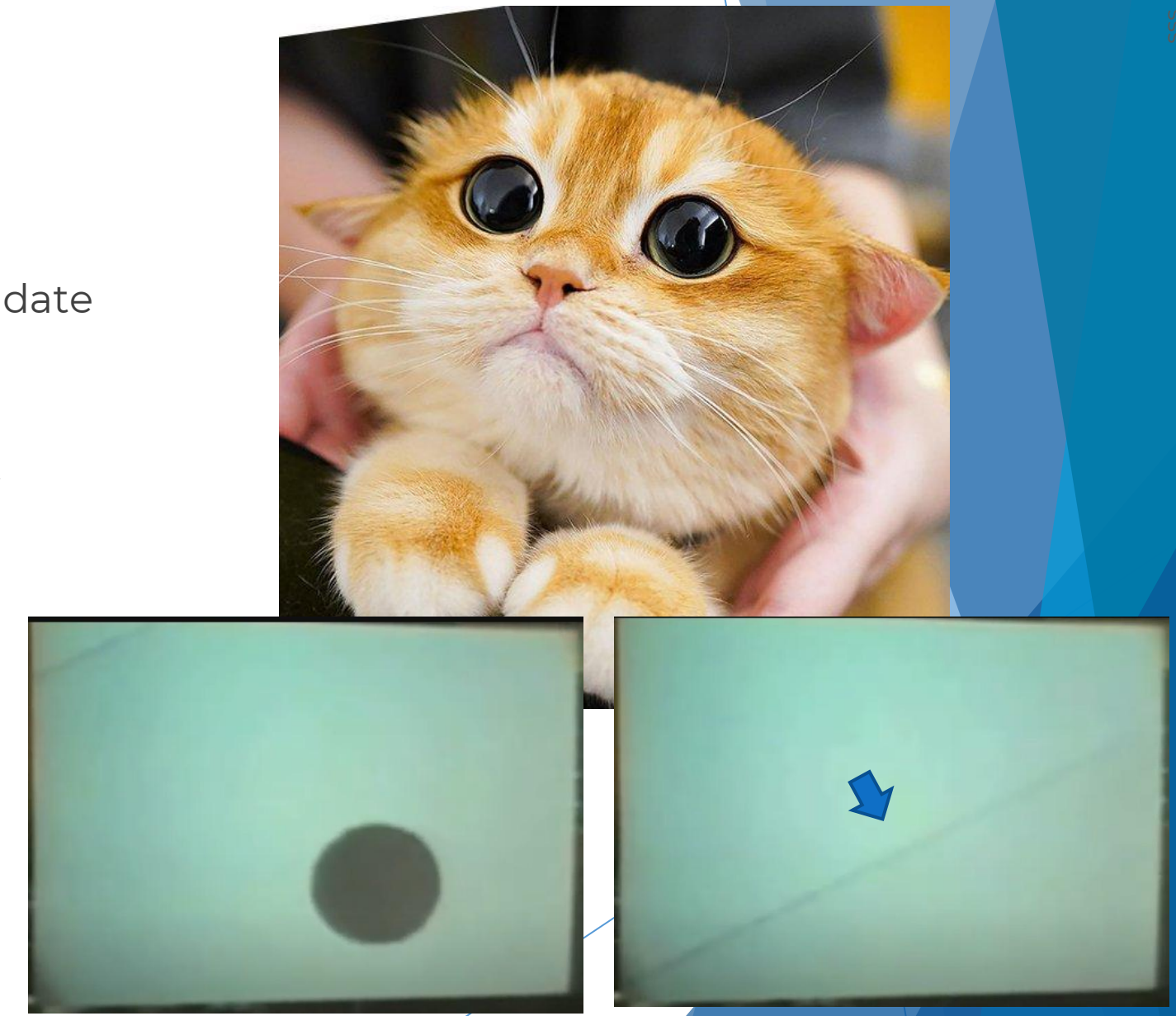
Computer Vision

Computer vision is the field of computer science that works on enabling computers to **identify and process objects in the same way that humans do**.

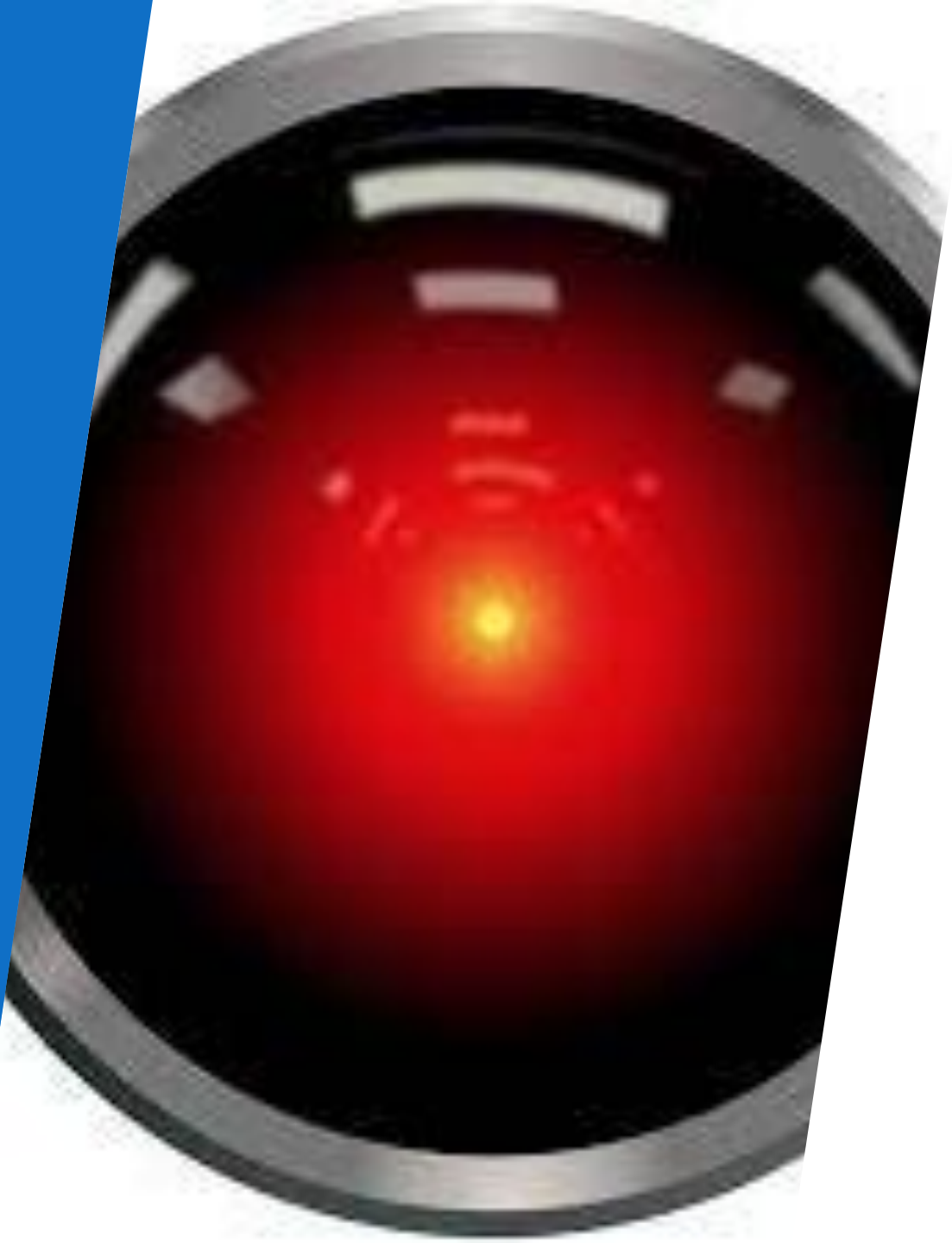
i.e., **make sense** of images and videos beyond a set of pixels or simple regions providing a **meaningful interpretation**

Some History

- The foundations of Computer Vision date back to the 1950s and 1960s
- Experiments with cats studying how neurons react to various stimuli
- Human vision discovered to be hierarchical
  - Edges, shapes, objects,...



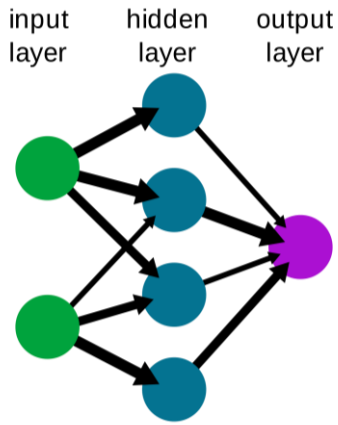
https://www.youtube.com/watch?v=IOHayh06LJ4



Some History

- The field of artificial intelligence is founded in Dartmouth College in 1956
- **Machines as intelligent as human beings** predicted **in a generation time**
- In **1966** Sussman and Minsky work on linking a camera to a computer and **having the computer describe what it saw**... something **ongoing, nowadays**...

Some History

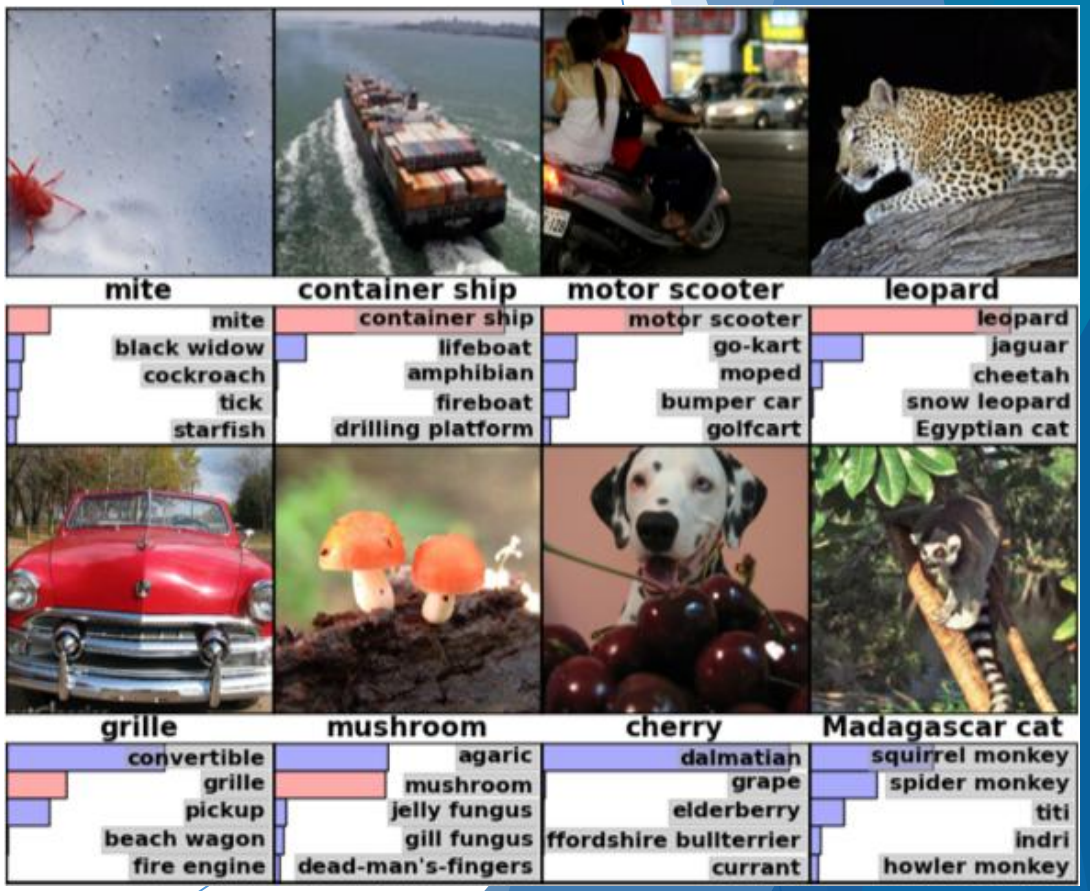


Neural Networks gain momentum in the 1980s

Cornerstone at ImageNet Large Scale Recognition Challenge (ILSVRC) 2012 with the introduction of a deep neural network: **AlexNet** (error rate: 15.3%; second place was 26.2%)

AlexNet considered 1.2 million images to train

Since then, error rates fell to **just a few percent**

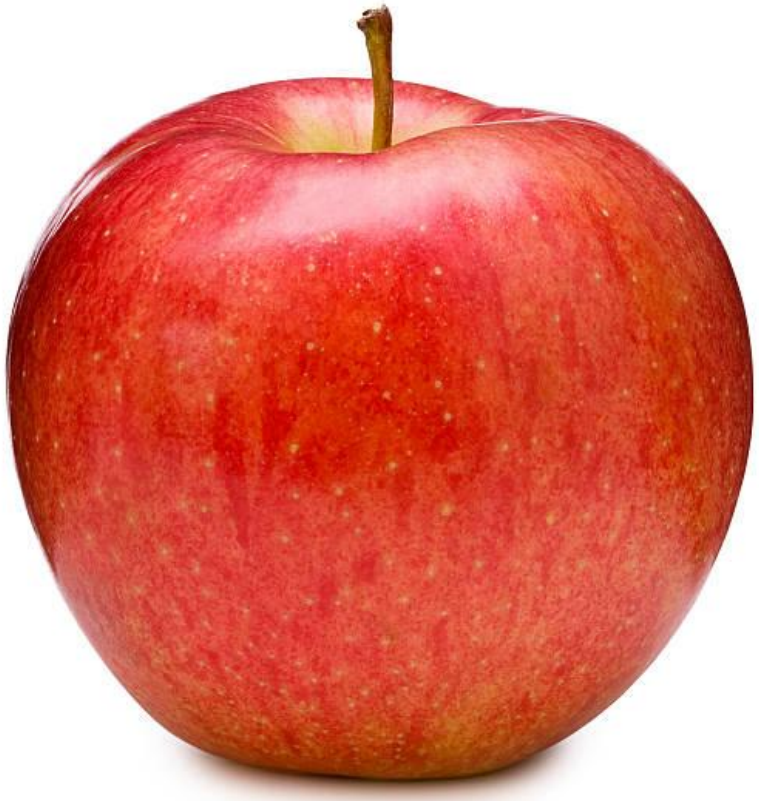


https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951

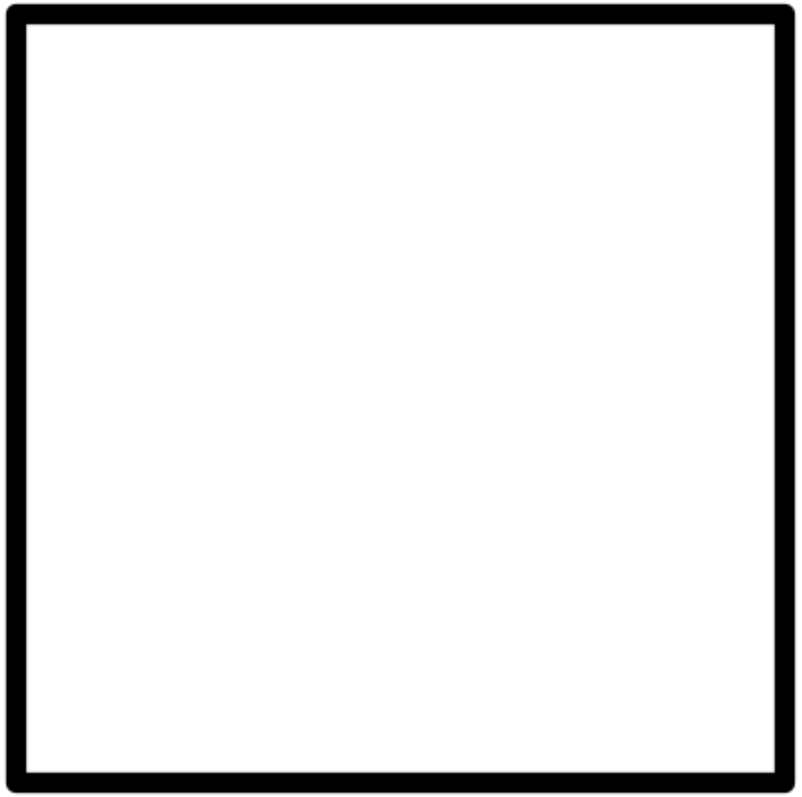
How do humans recognize objects?

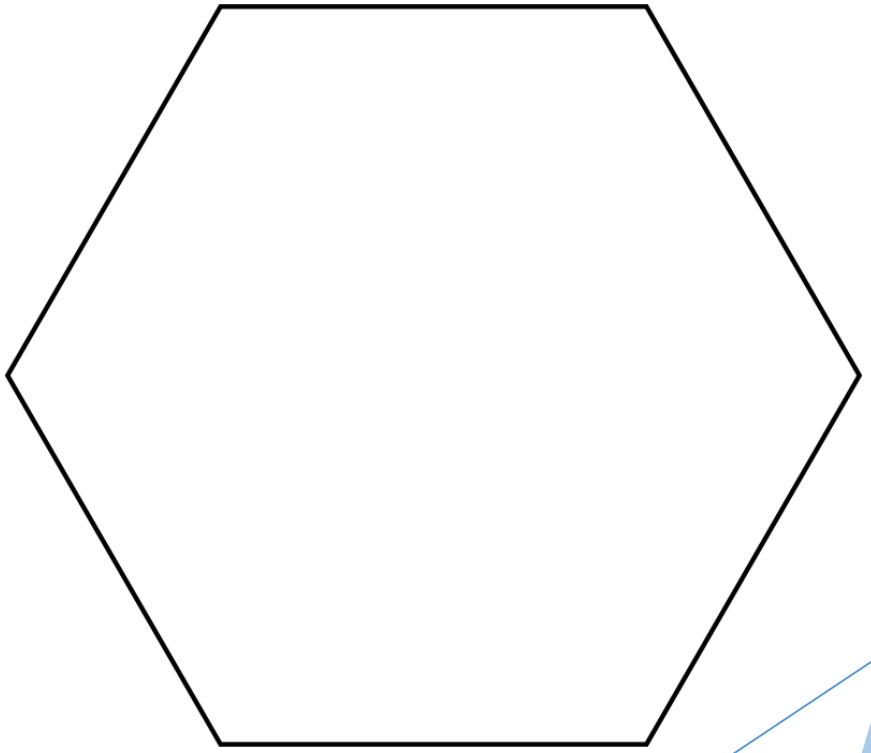
How do we make the distinction?

Without going into visual perception…



How do we make the distinction?





How do we make the distinction?

We Recognize Patterns

So, we – more or less – easily **find patterns** that help us identify different objects at different semantic levels (primate, person, old man, Samuel)

For computer vision, **how can we approach the problem in a similar fashion**?

Image Features

Features are a form of representing **object characteristics** and patterns that may allow identifying it

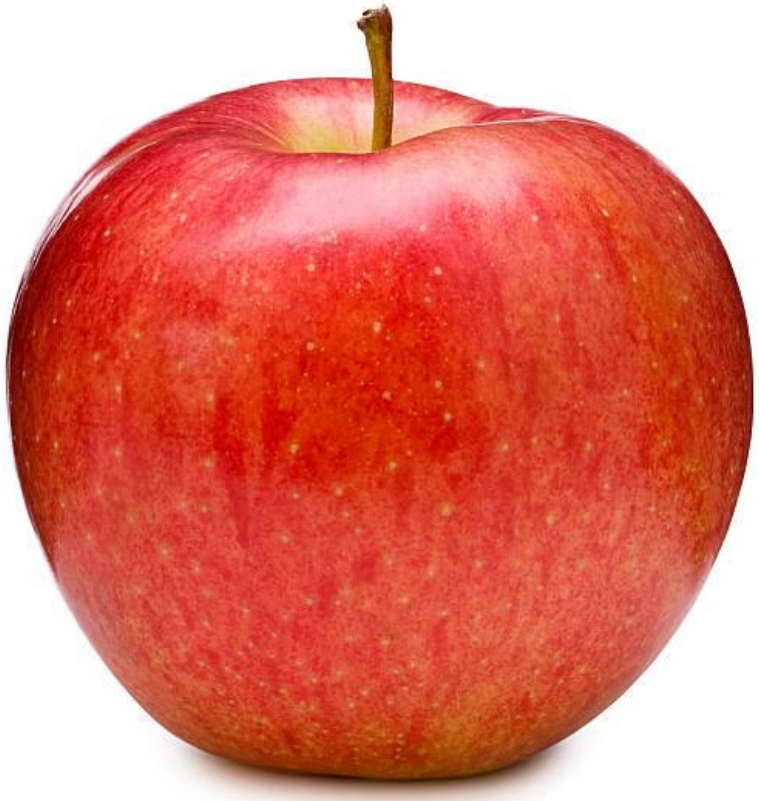
They can be local or global

Color, edges, corners, luminance, gradient,...

Often considered in sets (e.g., **feature vectors**)

How do we make the distinction?

{color = red}



{color = orange}

How do we make the distinction?

{color = red, +??}

{color = orange, +??}

More than one feature may be required to distinguish between objects

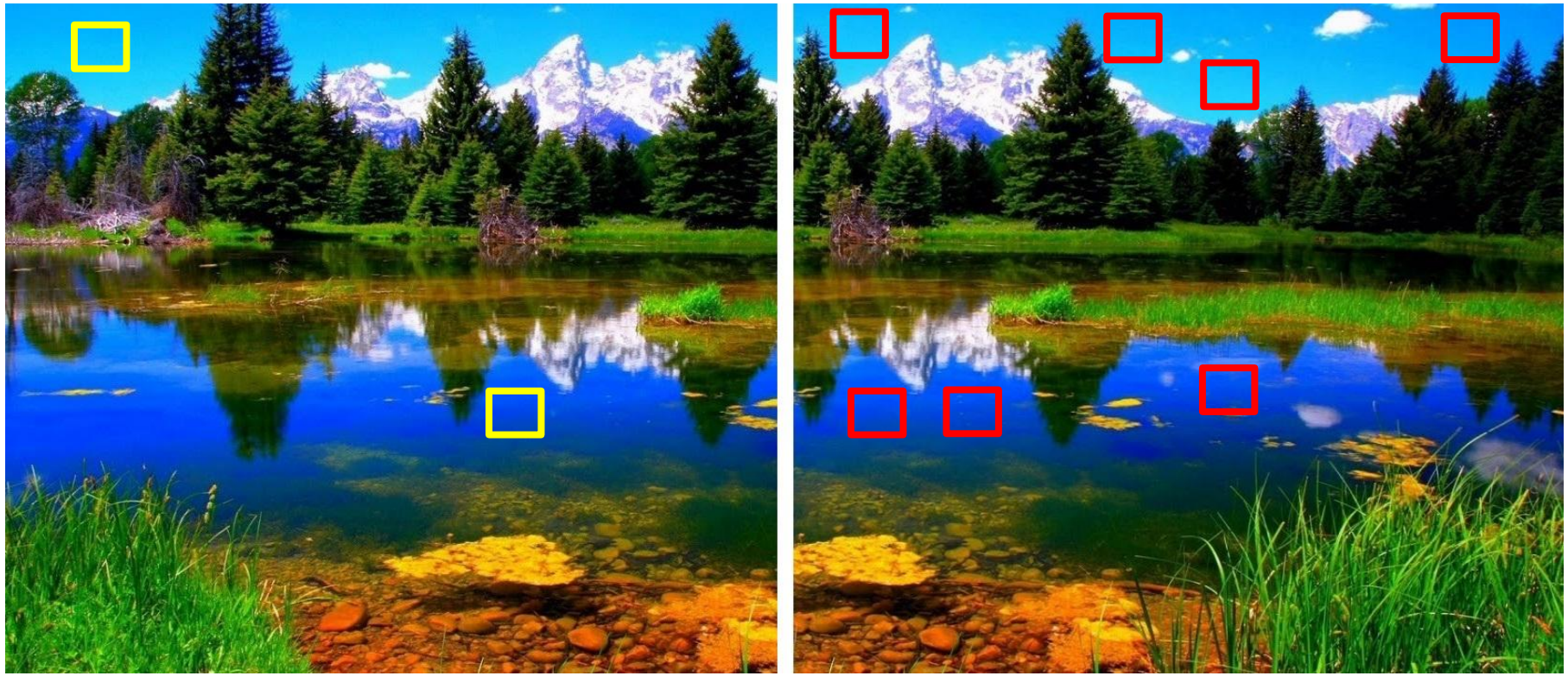
Matching Problem

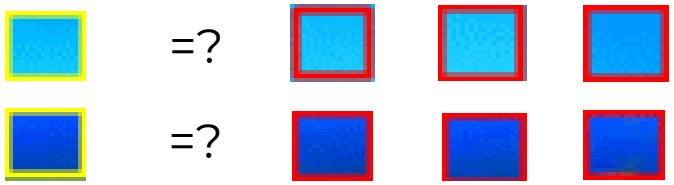
Matching Problem

How do we align the images to obtain the larger panorama?

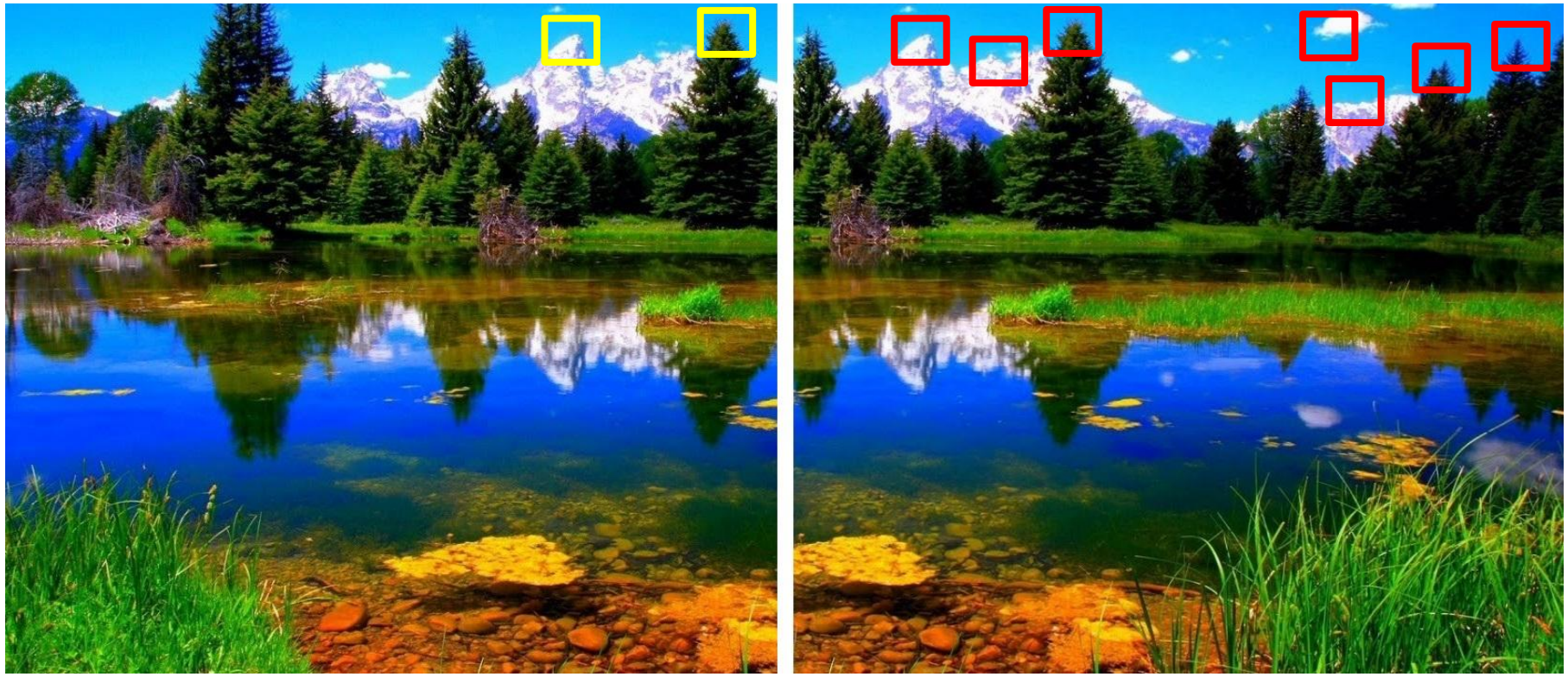
https://pylessons.com/OpenCV-image-stiching

Matching Problem





Matching Problem



Are these any better?

Corners

Junctions of contours

Generally, **less variable with changes of viewpoint**

Characterized by **large variations in the neighbourhood** around it

Feature Examples

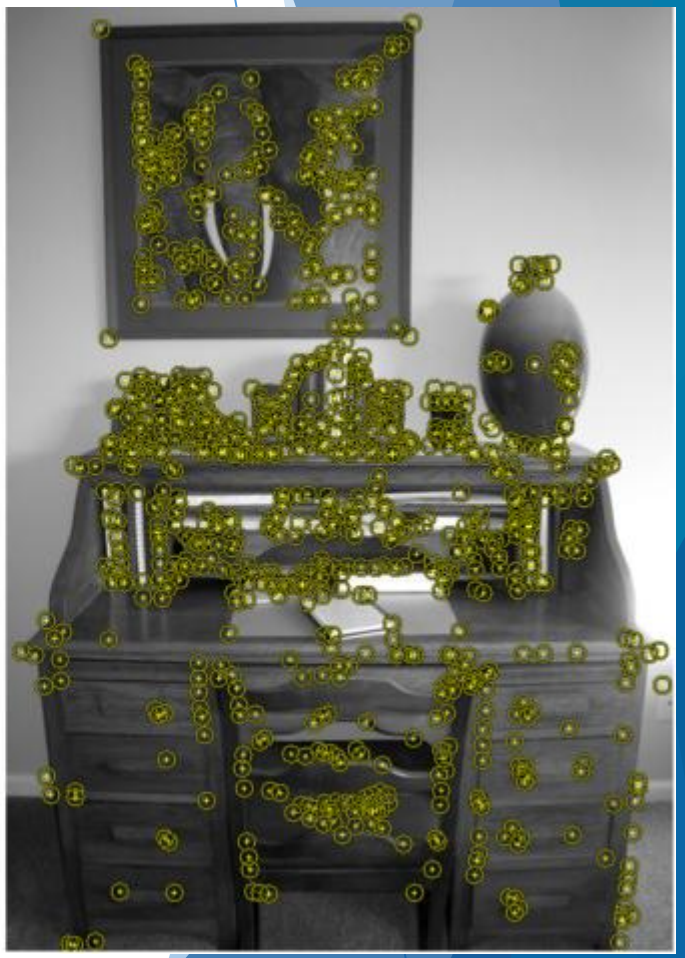
Corners

Harris Corner Detection

(or **Shi-Tomasi Corner Detector**)

What happens to the results if we rotate the image?

What happens if we scale the image?



Feature Examples

Scale-Invariance

Scale-Invariant Feature Transform (SIFT)

- Checks different image scales using gaussian filtering to determine **keypoints**

Speed-up Robust Features (SURF)

- Faster version of SIFT

Overall idea is to determine **keypoints** and compute a **descriptor** at each keypoint considering the neighbourhood

Feature Examples

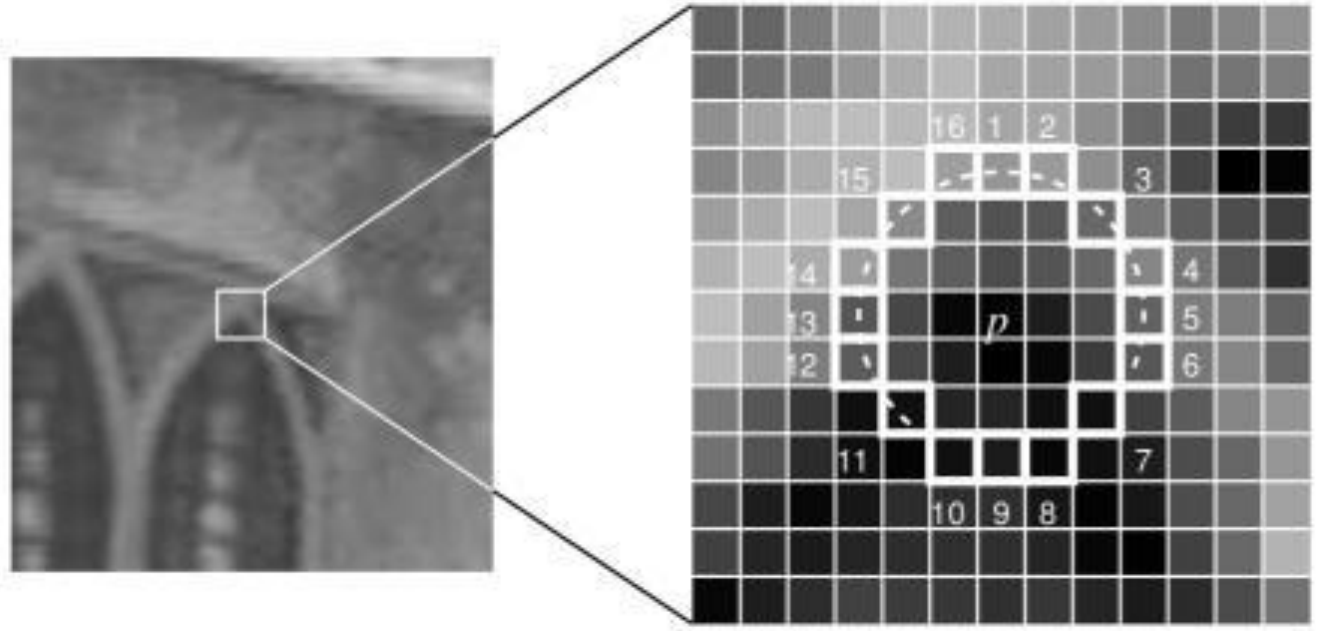
Faster approaches

Features from Accelerated Segment Test (**FAST**)

Binary Robust Independent Elementary Features (**BRIEF**)

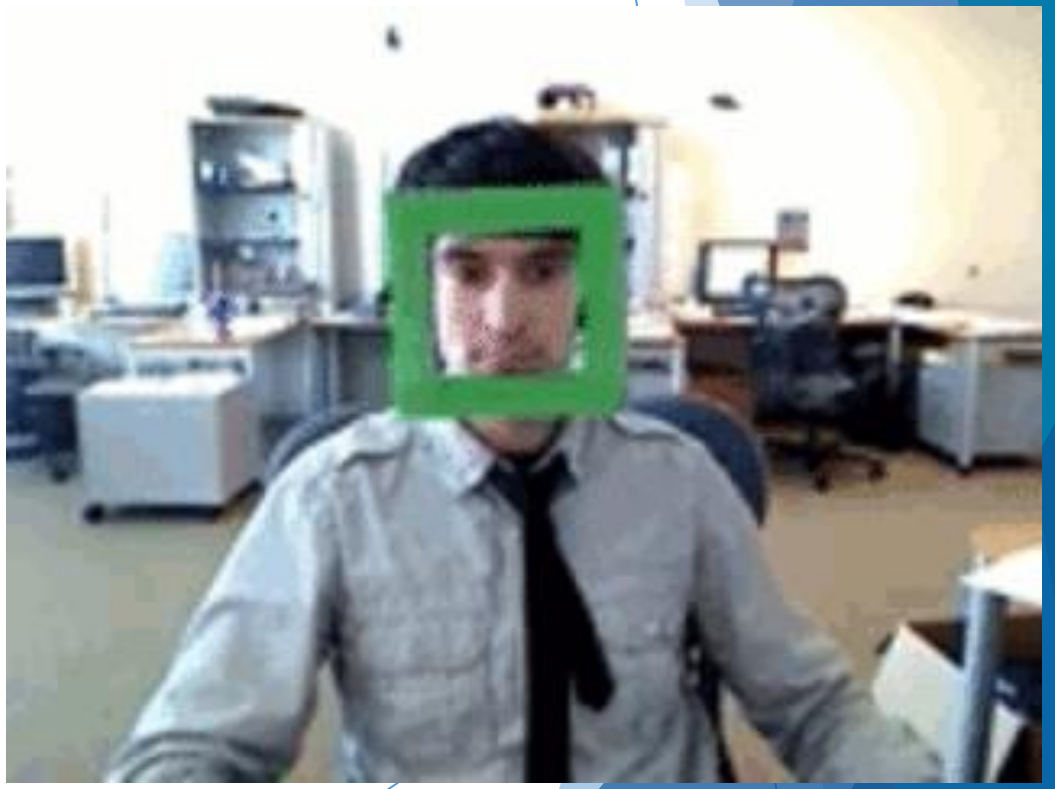
Oriented FAST and Rotated BRIEF (**ORB**)

Overall idea is to determine **keypoints** and compute a **descriptor** at each keypoint considering the neighbourhood



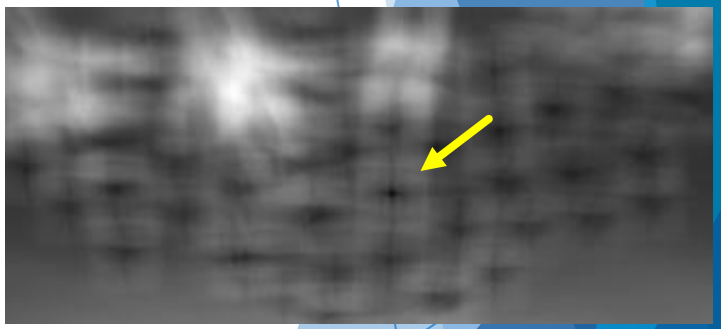
Object tracking

How do we follow an object in a video?



https://learnopencv.com/wp-content/uploads/2017/02/real-time-face-tracking.gif

Object tracking
template matching



Slide template over image and compute distance

Find point (s) of least distance

https://towardsdatascience.com/object-detection-on-python-using-template-matching-ab4243a0ca62

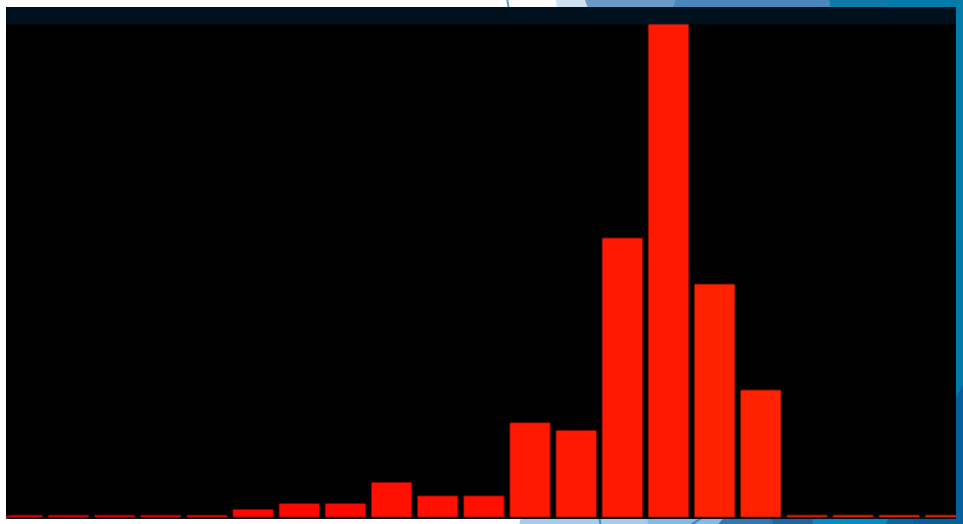
Object tracking
histogram matching

Instead of using the pixels as template, one can use the histogram

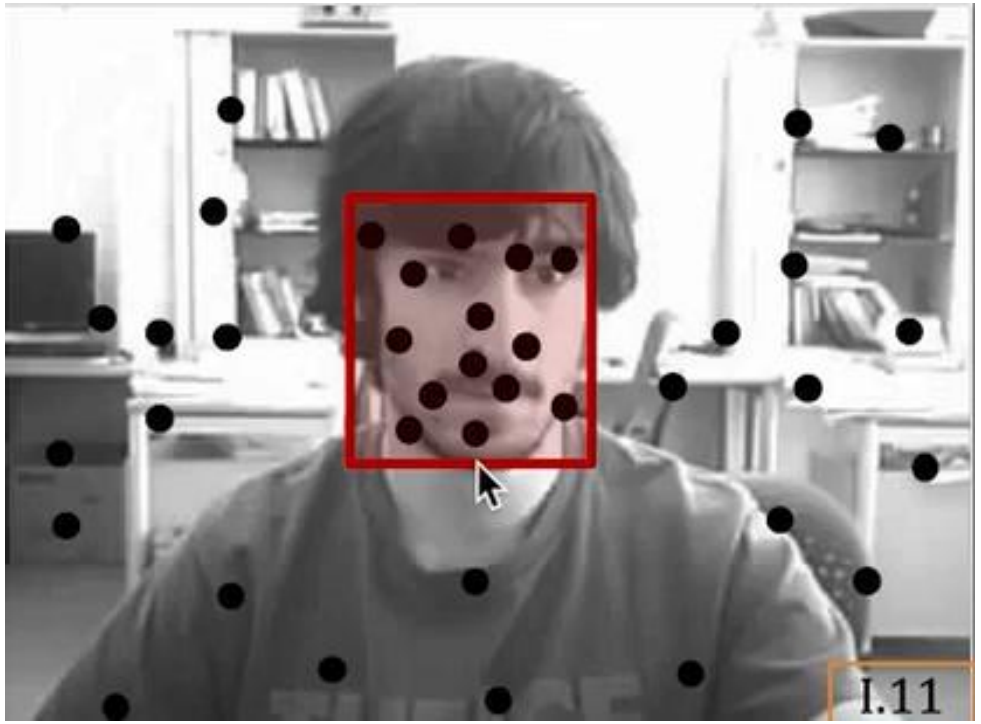
- Slightly more immune to large scale changes or some rotations

Sliding window over image computing histograms and their distance to template

If there are similar objects/background (i.e., with similar histogram) matching will be tricky or varying



Object tracking
Feature detection

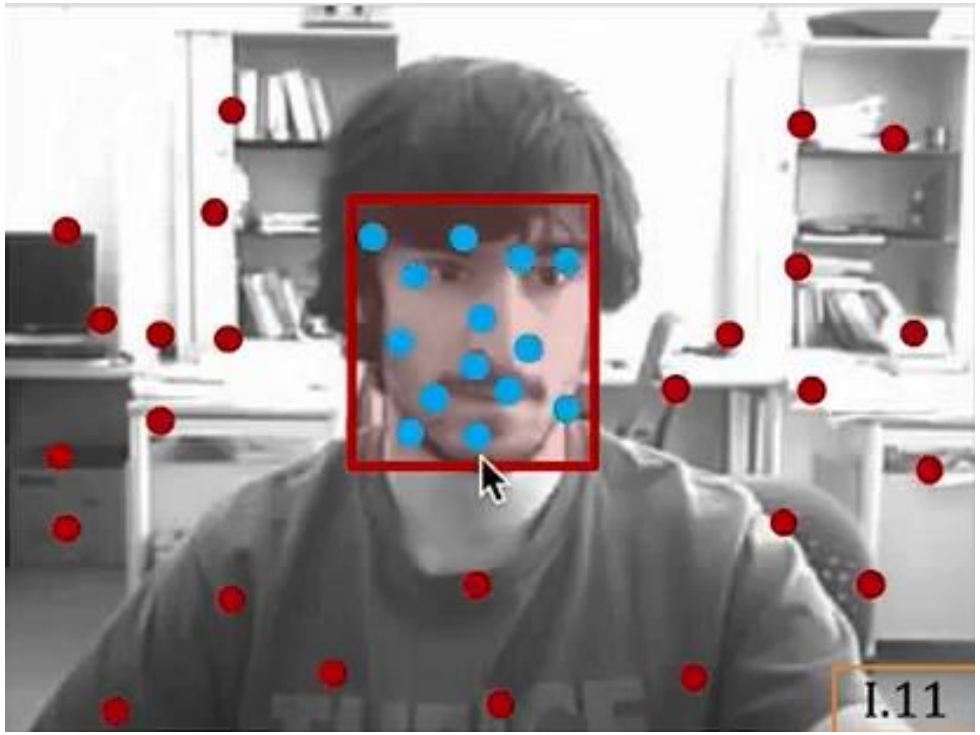


Select object (e.g., manually, or face detection)

Compute features (e.g., SIFT or ORB)

[Credit: Prof. Shree K Nayar]

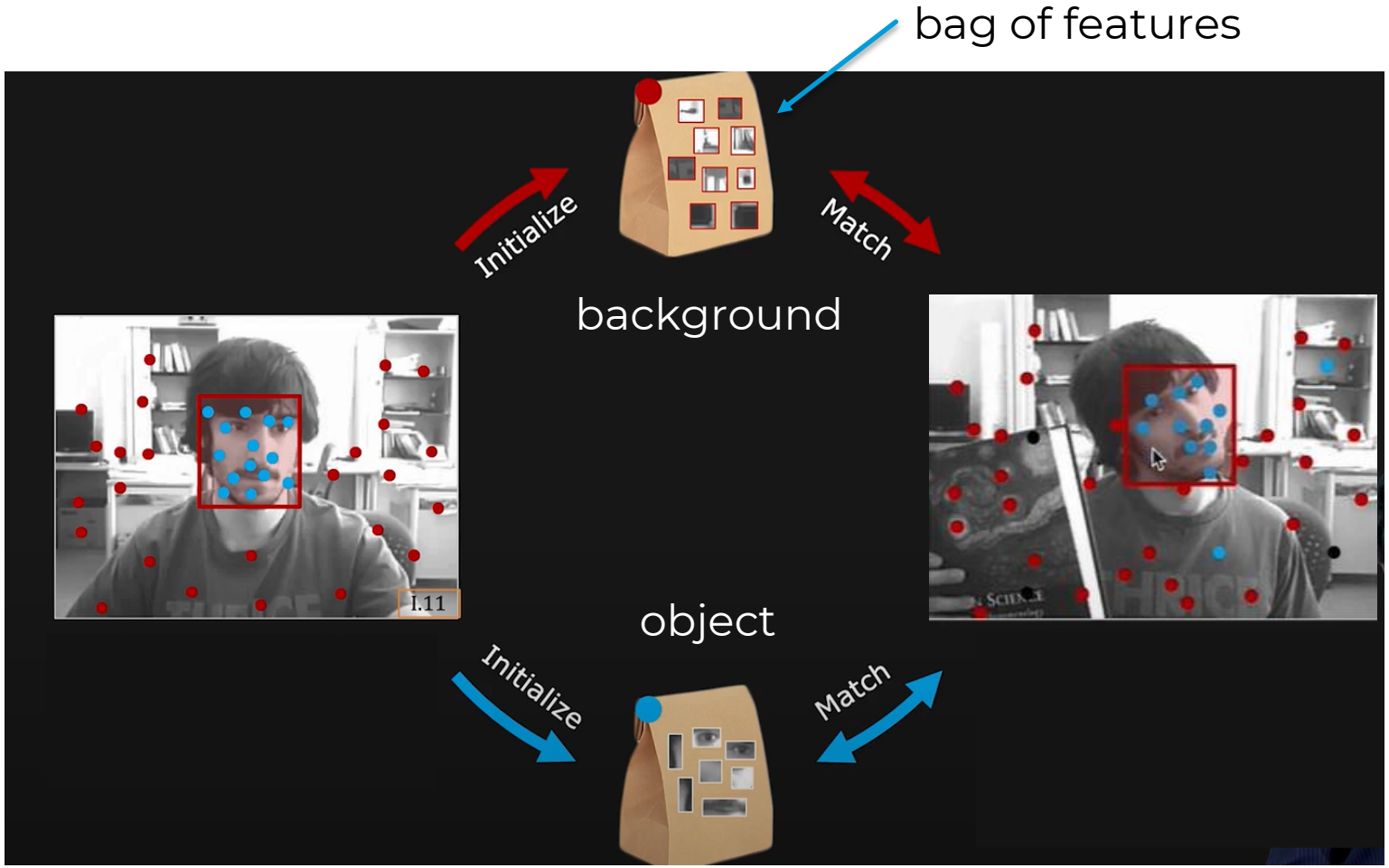
Object tracking
Feature detection



Find all features for object and for background

[Credit: Prof. Shree K Nayar]

Object tracking
Feature detection



Features selected as **background** and **object**

Features computed for new frame

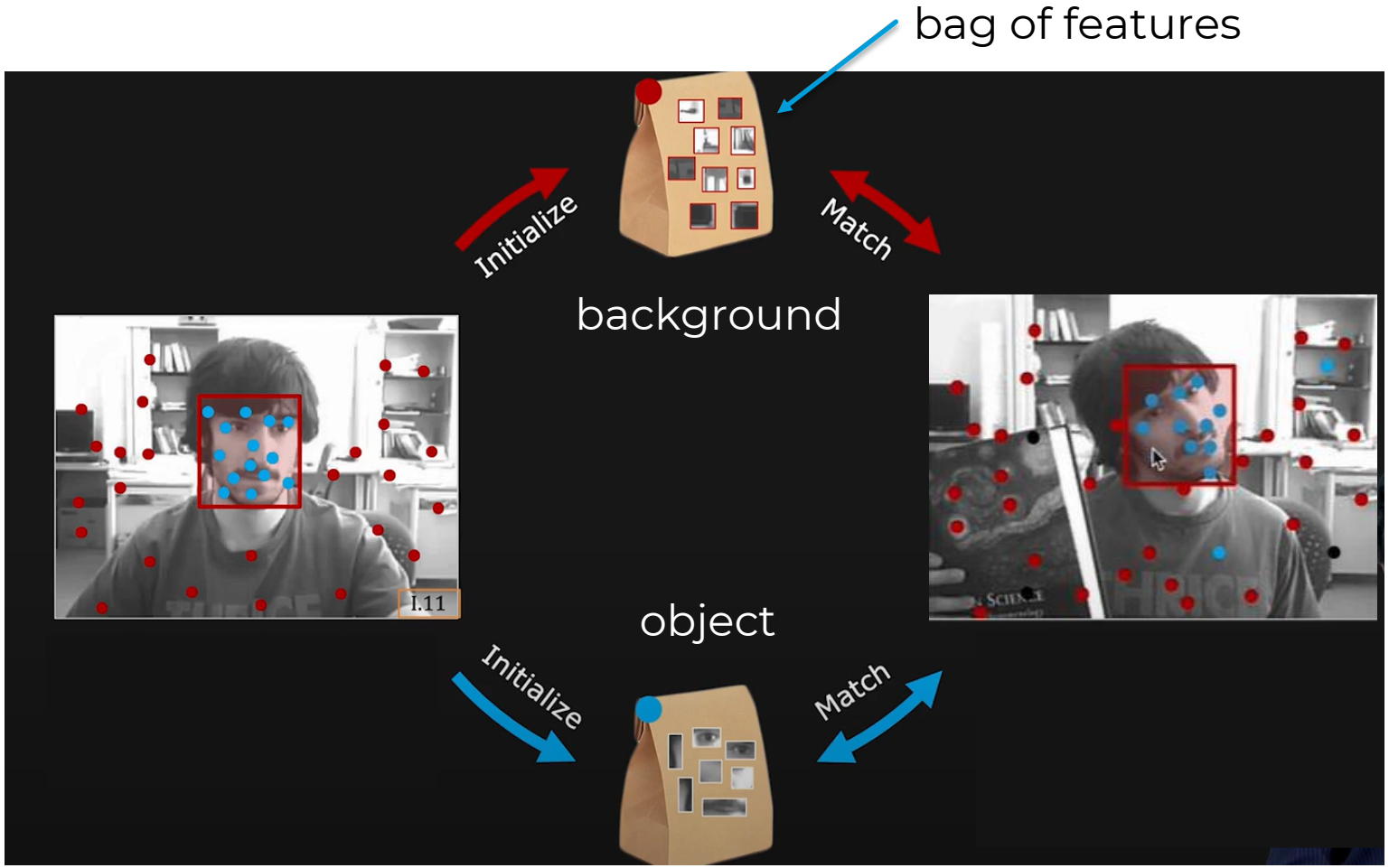
New features compared with bags

“Closest bag” sets type of feature

Object window encompassing the **most object features and less background**

[Credit: Prof. Shree K Nayar]

Object tracking
Feature detection



Since object changed position and orientation, **refresh object and background bags of features** for next frame

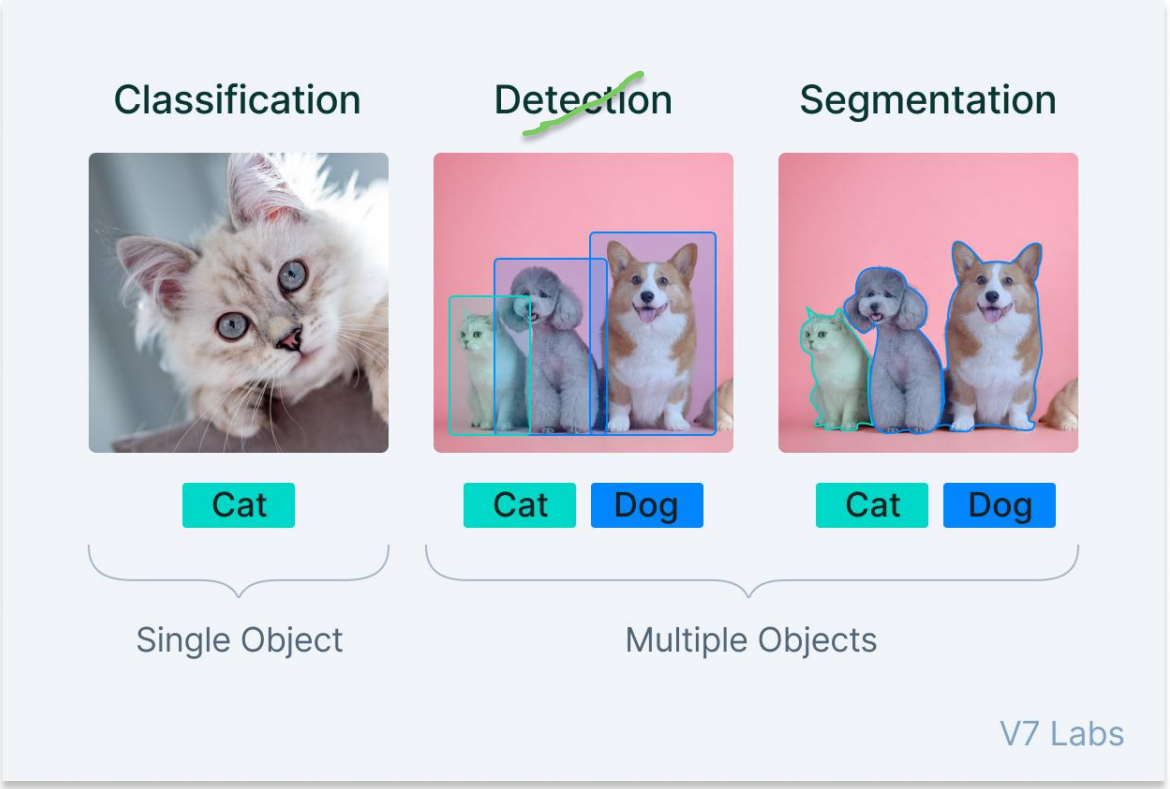
[Credit: Prof. Shree K Nayar]

Approaches to Computer Vision Implementation

Early days, machine learning, deep learning

Common Computer Vision Tasks

- **Object Classification**: What broad category of object is in this photograph?
- **Object Identification**: Which type of a given object is in this photograph?
- **Object Detection**: Where are the objects in the photograph?
- **Object Segmentation**: What pixels belong to the object in the image?
- **Object Recognition**: What objects are in this photograph and where are they?
- …..



Early days

Database of **labelled objects** to detect

Annotate the **database** with features judged relevant

Get a new image and **compute the same features**

Define criteria (algorithm) to check if it matches existing image (e.g., distance between features)

Most aspects manually defined by developer and the computer is just a way to apply it

Machine Learning

A **database** is created and **labelled**

A **set of features** are defined

Method learns from the examples **what features work best** to distinguish between the classes

In this approach, **decision criteria (model) is optimized by the method**

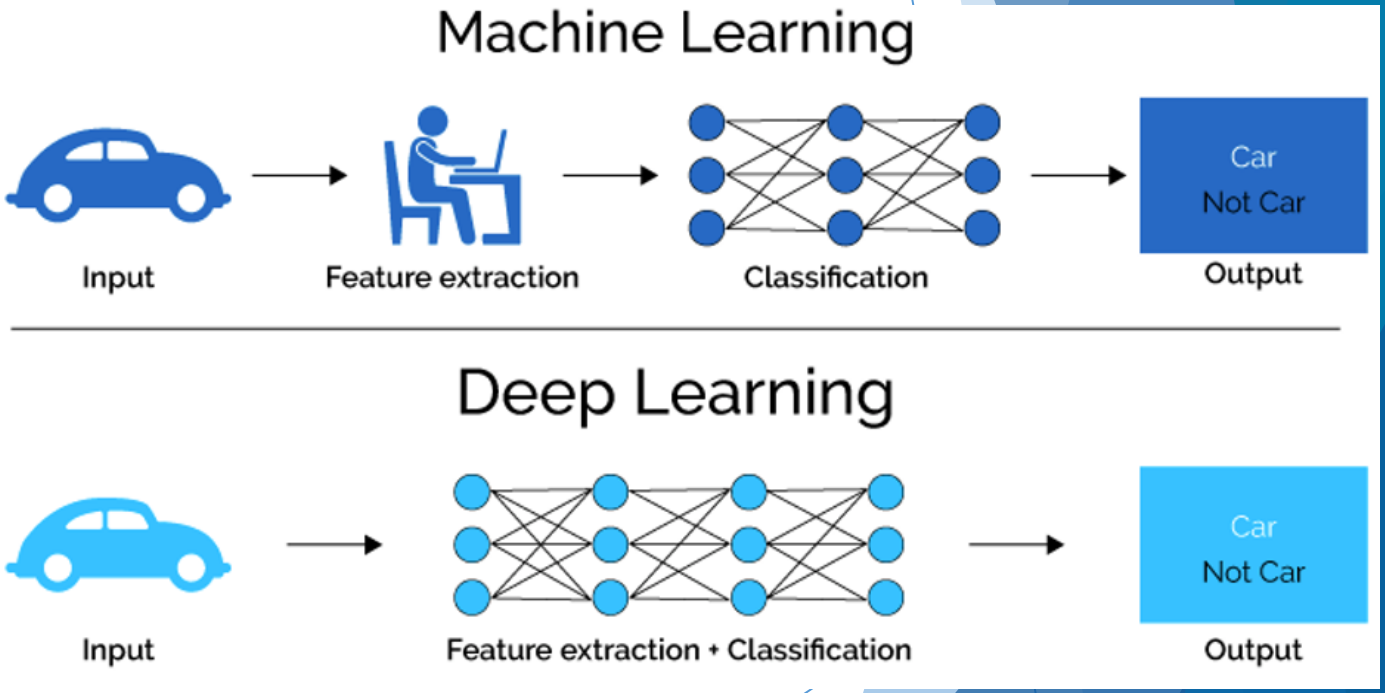
Deep Learning

Relies on neural networks

Solves a problem by going through a **large** set of labelled data

It finds patterns that allow distinguishing between the classes

Does not require defining the features or rules

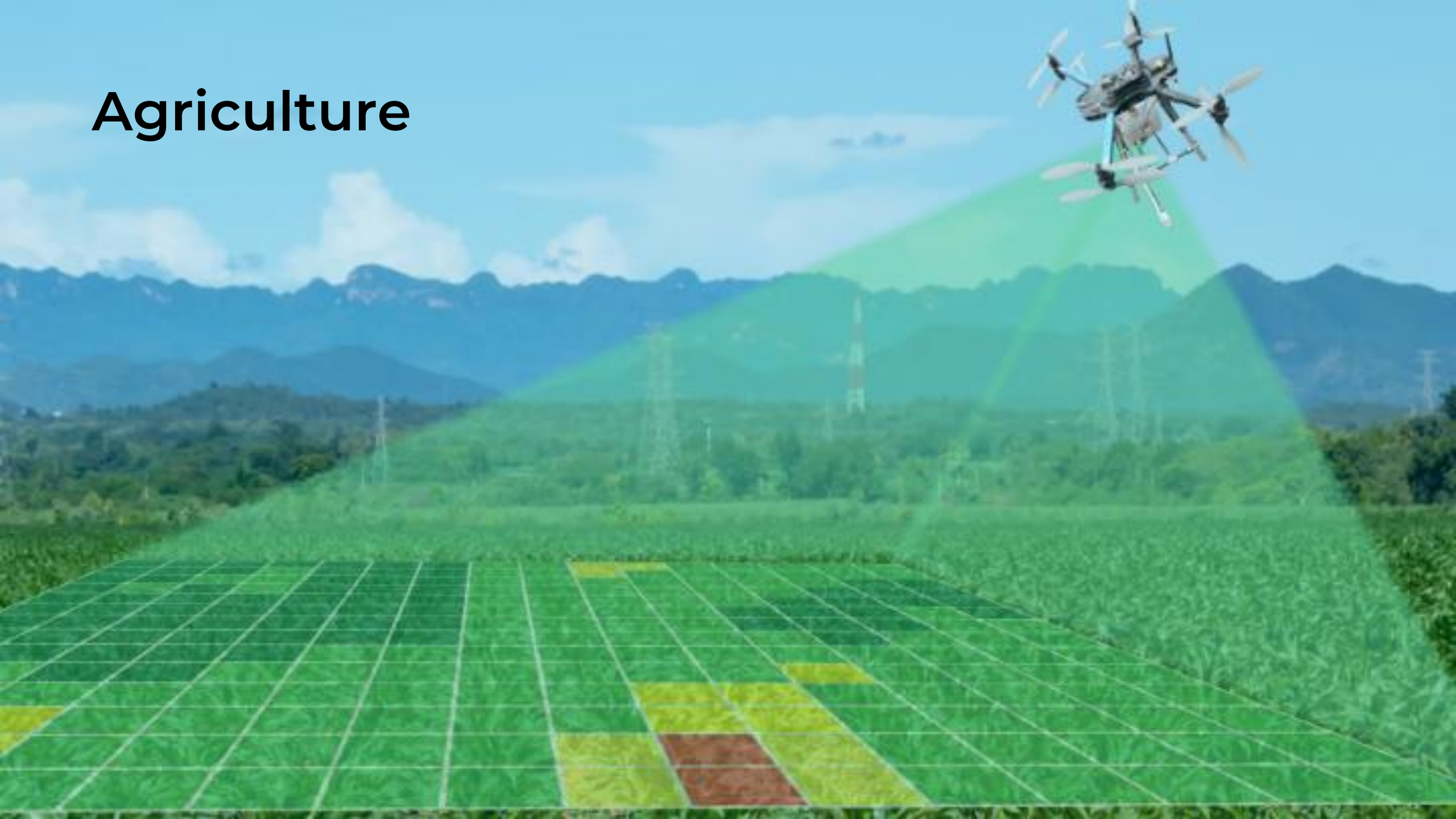


https://xd.adobe.com/ideas/principles/emerging-technology/what-is-computer-vision-how-does-it-work/


Applications of
Computer Vision


Facial Recognition
mobidev

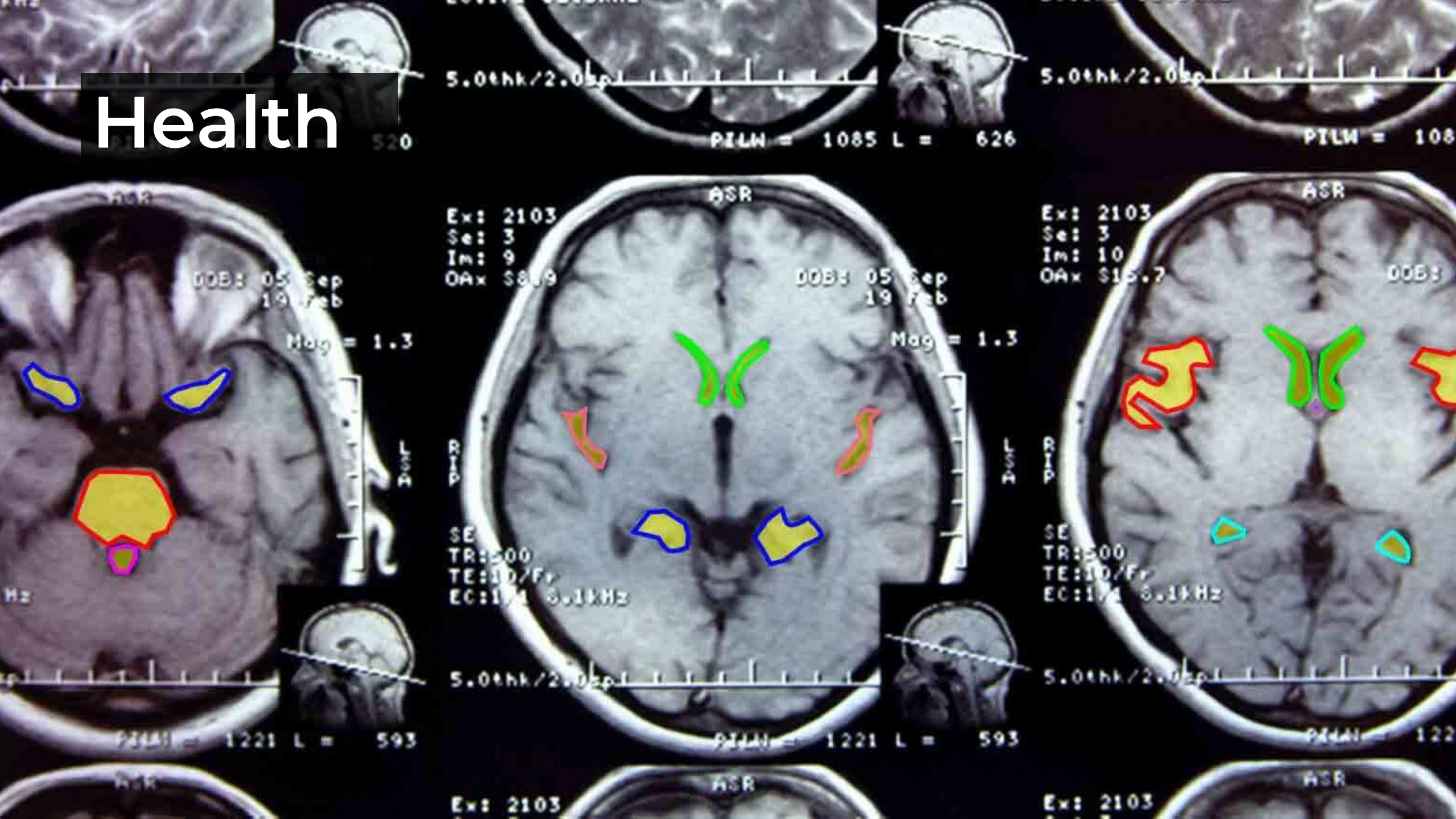
Agriculture



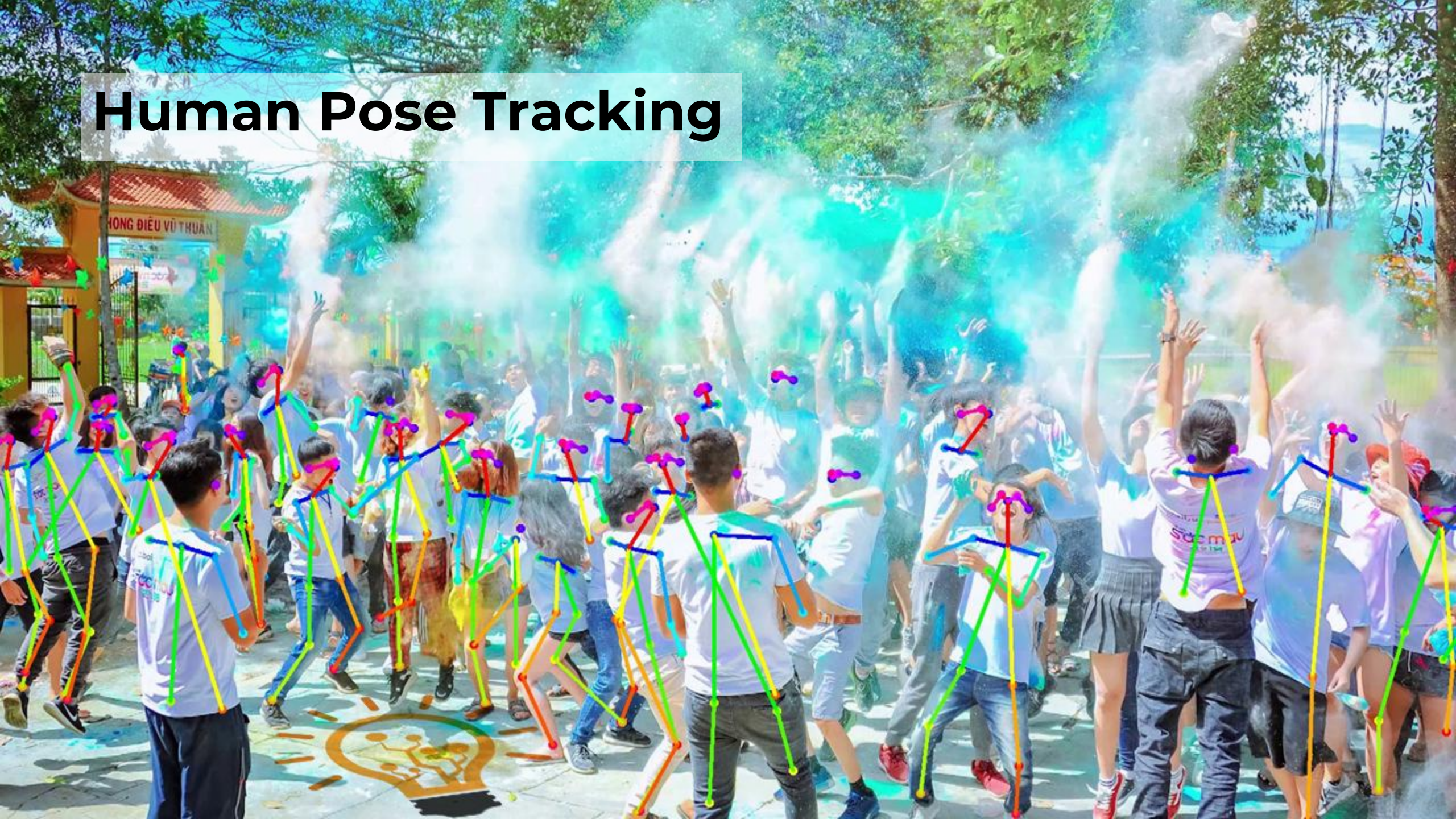
Autonomous Driving

Augmented Reality

https://industrywired.b-cdn.net/wp-content/uploads/2020/09/AR.jpeg


Health

Human Pose Tracking



Smart Cities

https://www.nvidia.com/content/dam/en-zz/Solutions/industries/smart-cities/nvidia-industry-ai-smart-cities-iva-2c50-l@2x.jpg

Off-the-Shelf Solutions

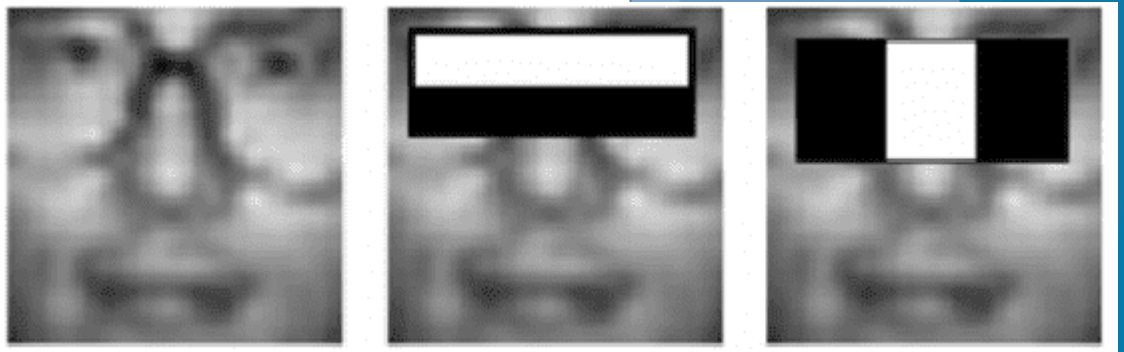
Off-the-Shelf Solutions

There are **several pre-trained models** for several types of objects

These can be applied directly to new images or video streams for object detection, identification, etc.

Remember: **pre-trained models detect a set of objects, not everything that we want**...

OpenCV
cascade classifiers



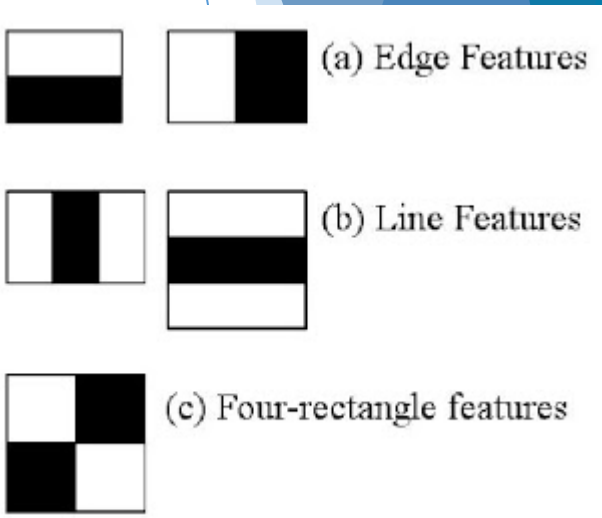
Features are like kernels applied to the hole image in all sizes and locations (24x24 image = 160000 features!)

During training this is reduced to around 6000 features

These are applied in cascade to classify image (e.g., if one fails, the image is discarded, and the process stops)

Several pre-trained classifiers (e.g., body, face, eyes)

Very fast!

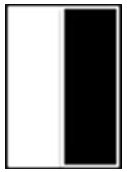


Viola & Jones (2002)

https://github.com/opencv/opencv/tree/master/data/haarcascades

Cascade Classifiers

Haar filters



Easy to compute, since it is just adding values

But it can be improved...

$$V_A[i,j] = \sum(\text{pixel intensities in white area}) - \sum(\text{pixels intensities in black area})$$

https://www.youtube.com/watch?v=5ceT8O3k6os&ab_channel=FirstPrinciplesofComputerVision

Computing Filter Response
Integral Image

Each position stores the sum of the image square with the right lower corner in that pixel

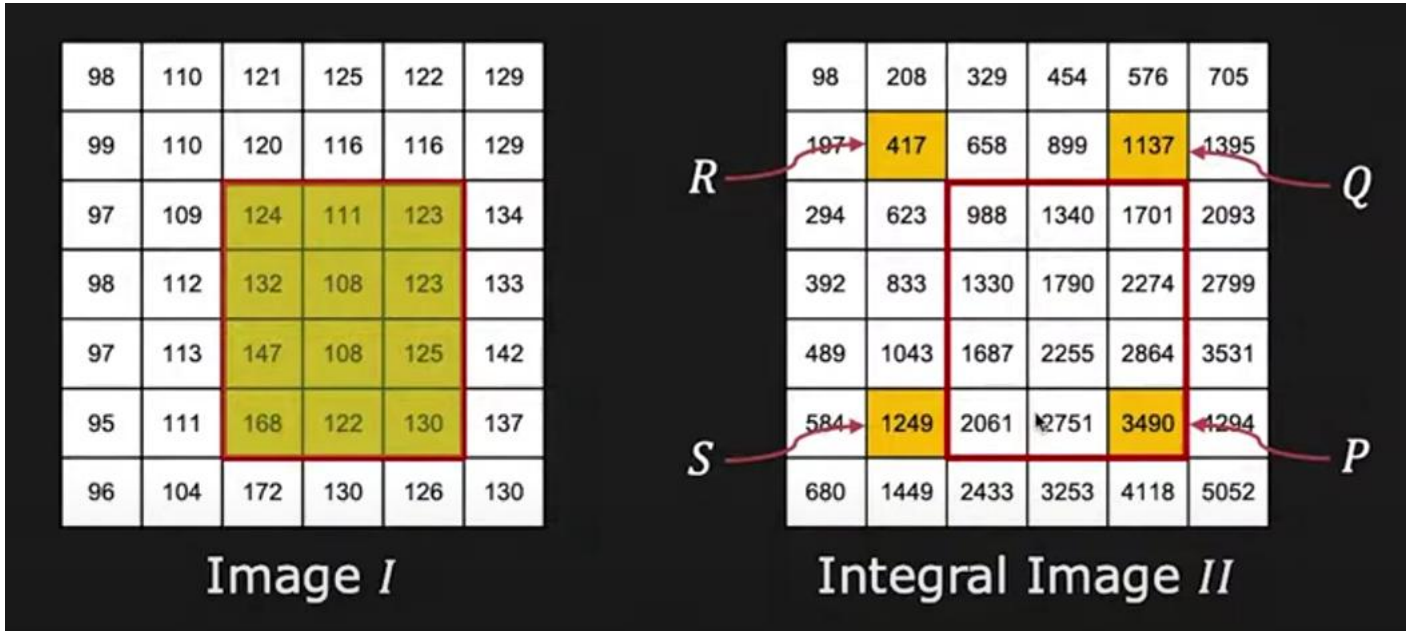
98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

https://www.youtube.com/watch?v=5ceT8O3k6os&ab_channel=FirstPrinciplesofComputerVision

Computing Filter Response

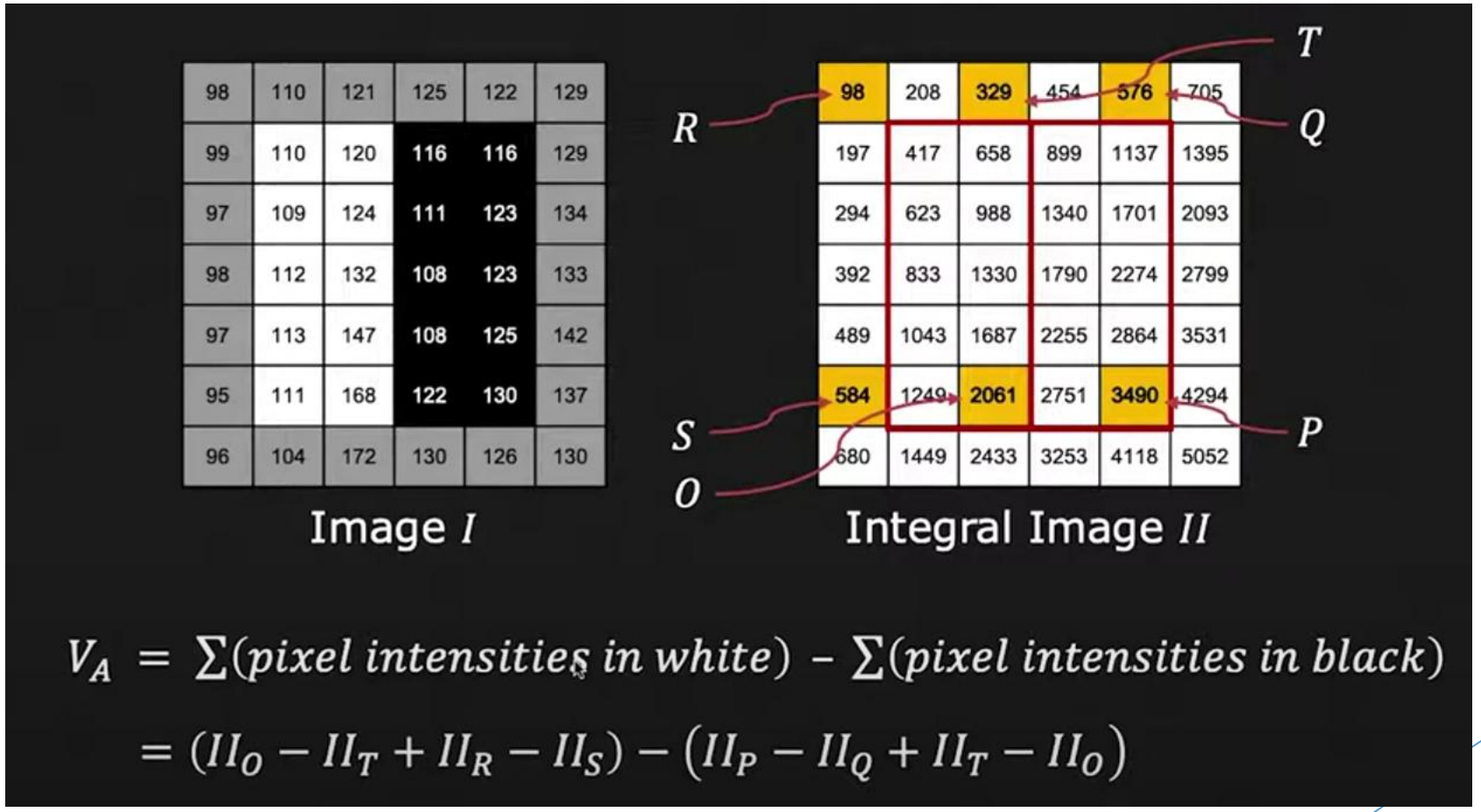
Integral Image



$$Sum = II_P - II_Q - II_S + II_R$$
$$= 3490 - 1137 - 1249 + 417 = 1521$$

https://www.youtube.com/watch?v=5ceT8O3k6os&ab_channel=FirstPrinciplesofComputerVision

Computing Filter Response

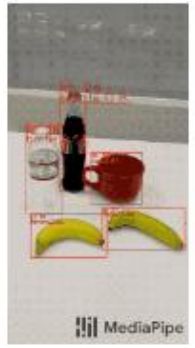


https://www.youtube.com/watch?v=5ceT8O3k6os&ab_channel=FirstPrinciplesofComputerVision

You Only Look Once (YOLO)

MediaPipe

Face Detection	Face Mesh	Iris	Hands	Pose	Holistic
					

Hair Segmentation	Object Detection	Box Tracking	Instant Motion Tracking	Objectron	KNIFT
					

Bibliography

- Learning OpenCV 4 Computer Vision with Python 3
- Abhinav Dadhich, “Practical Computer Vision”, Packt Publishing, 2018

Hands On!