INTERACTION DESIGN

COMPUTER VISION

Bits & Atoms IV

COMPUTER VISION

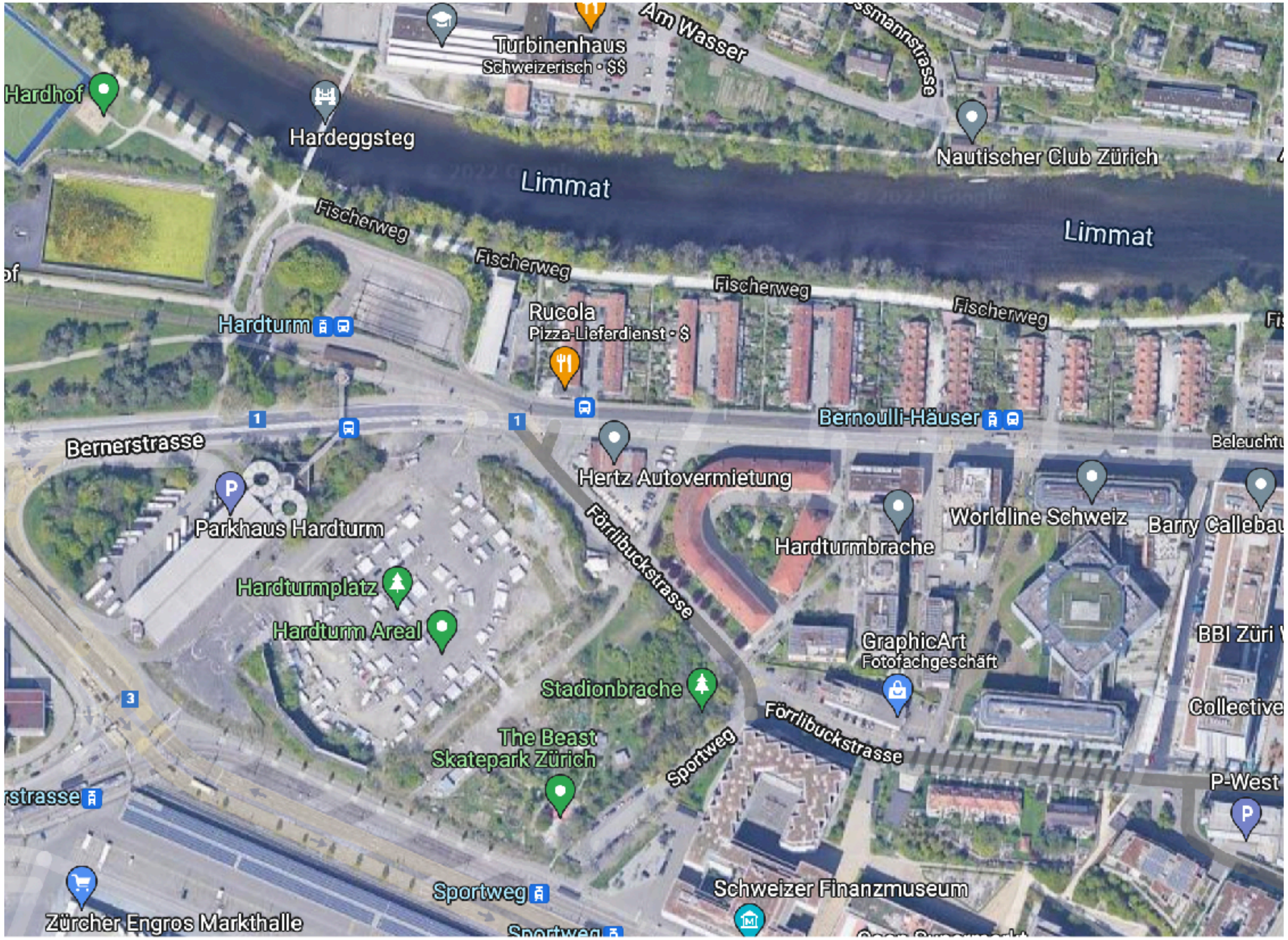
“Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs”

https://en.wikipedia.org/wiki/Computer_vision

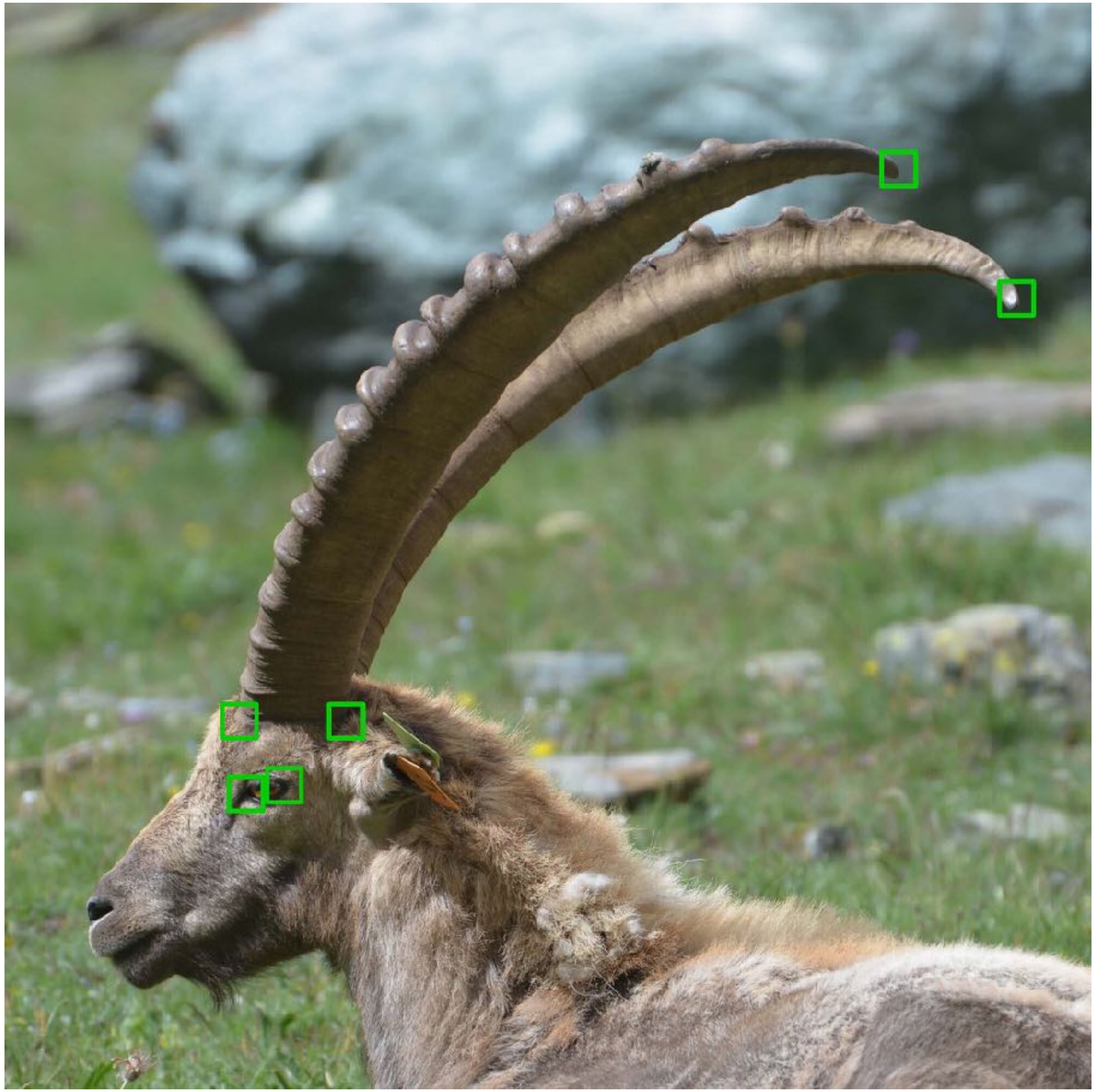
Why is computer vision relevant for spatial interaction?

- Automatic analysis/action
- Contextual information
- Understanding of 3D space on 2d projection

COMPUTER VISION



COMPUTER VISION



COMPUTER VISION

To get the most out of image data, we need computers to "*see*" an image and understand the content.

COMPUTER VISION

A person can describe the content of a photograph they have seen once.

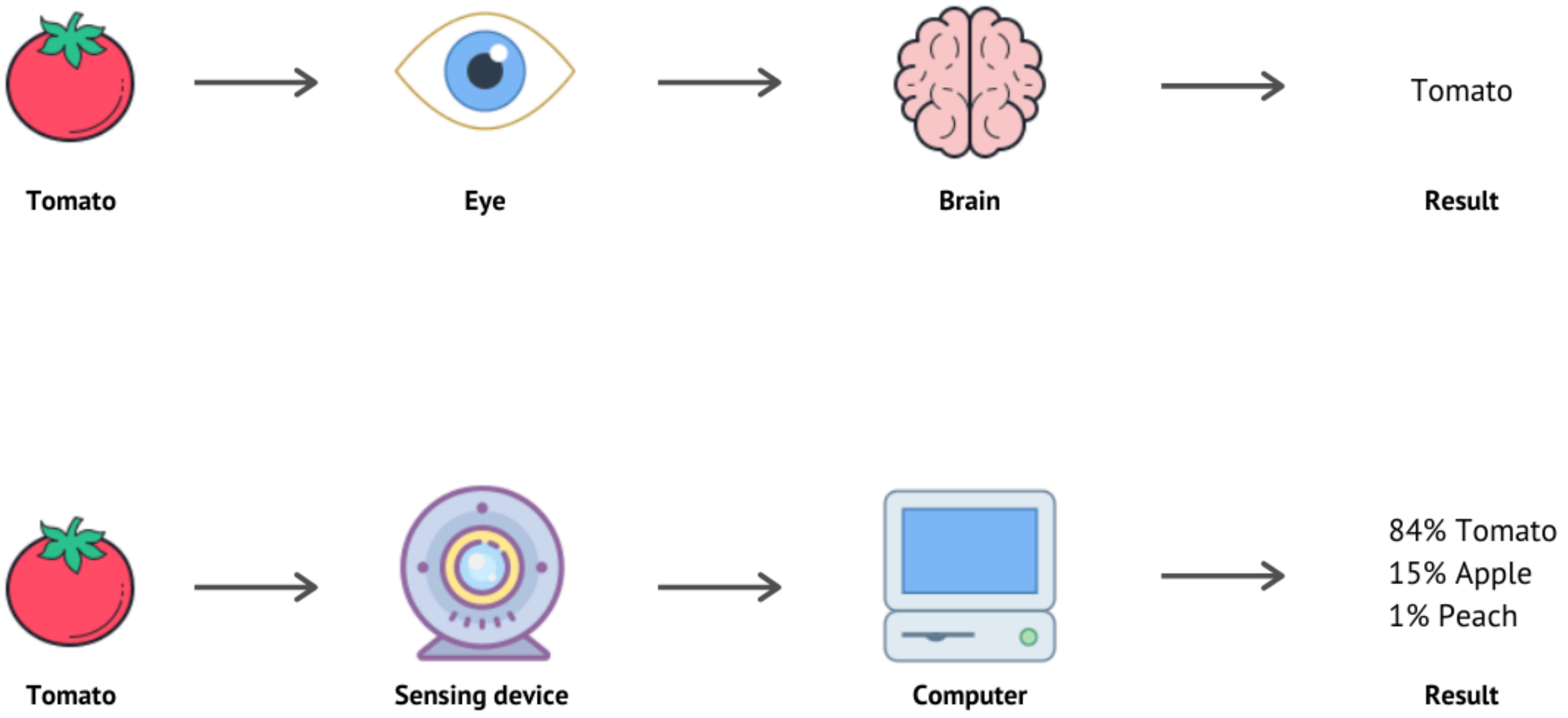
A person can summarise a video that they have only seen once.

A person can recognise a face that they have only seen once before.

This is a trivial problem for a human, but not for a machine.

COMPUTER VISION

Human Vision vs Computer Vision



COMPUTER VISION

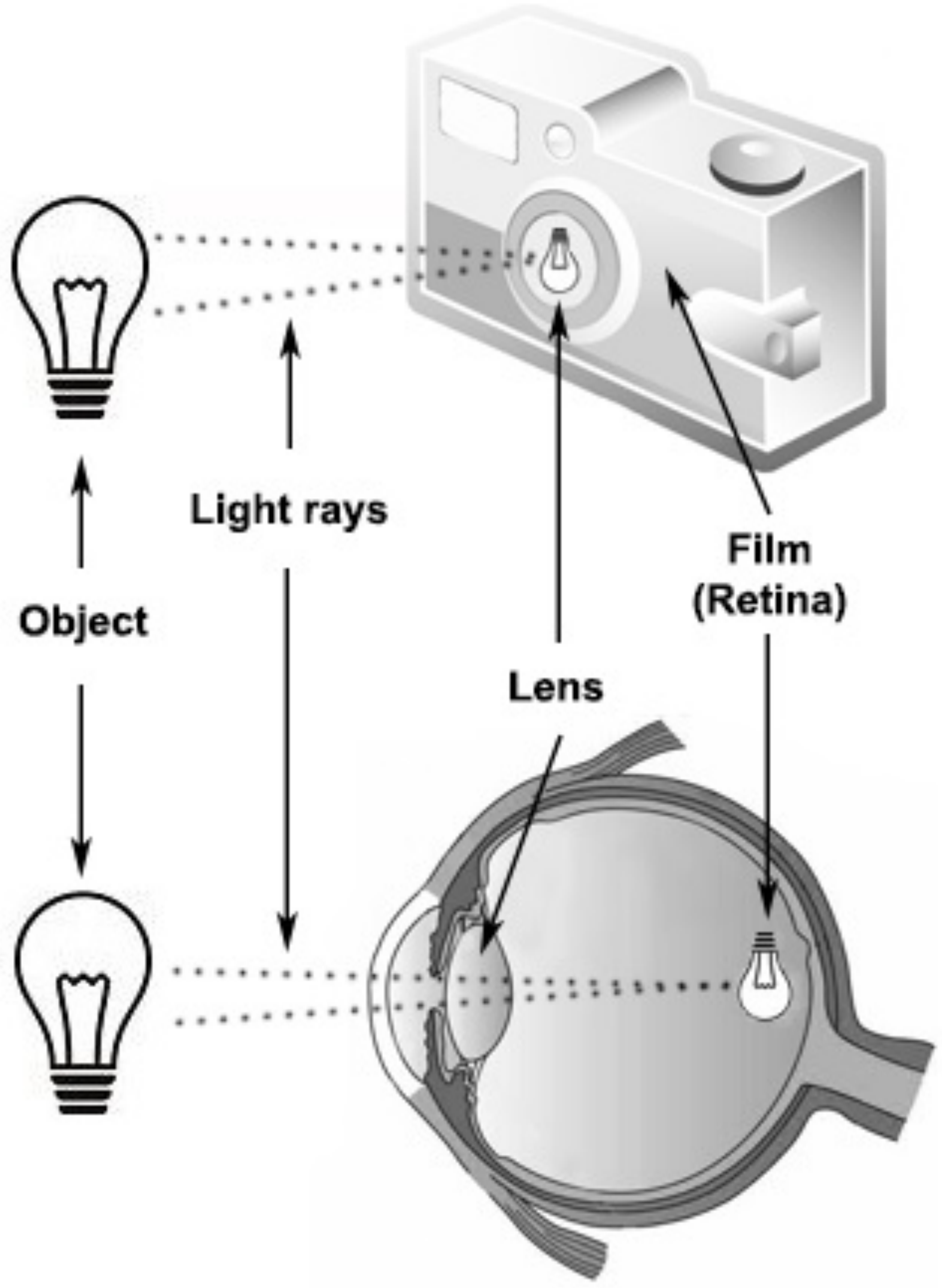
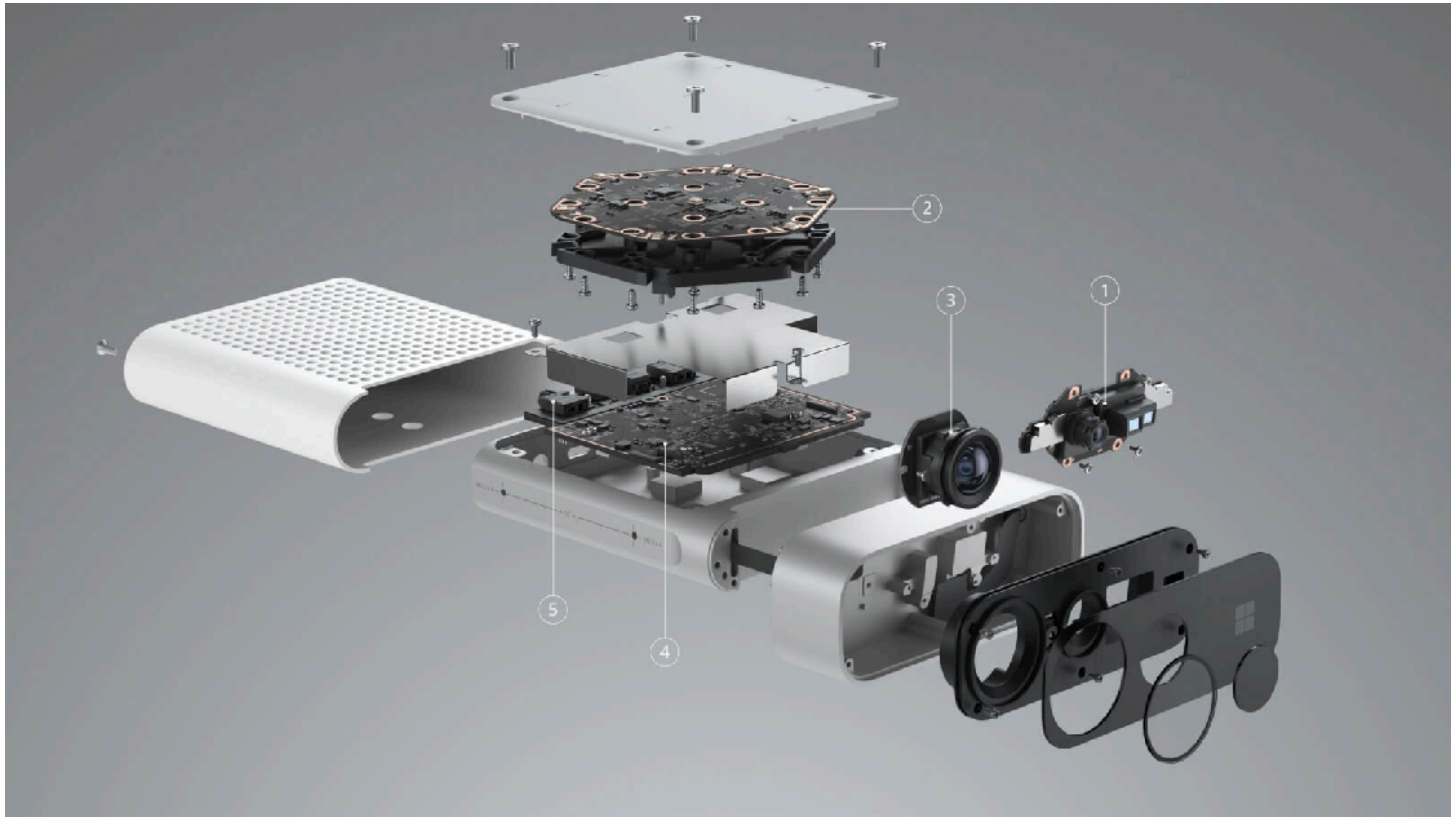
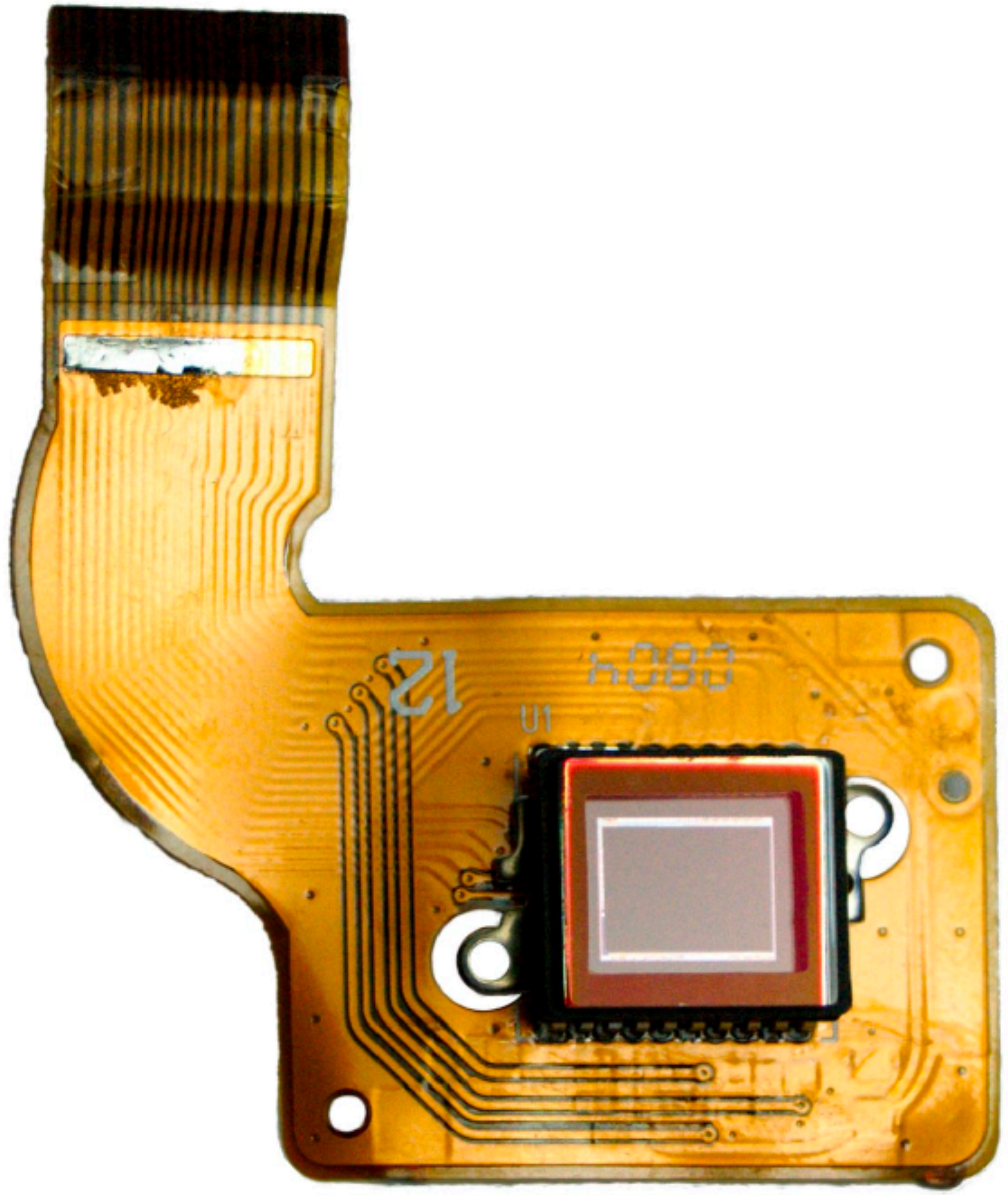


Image source unknown

COMPUTER VISION

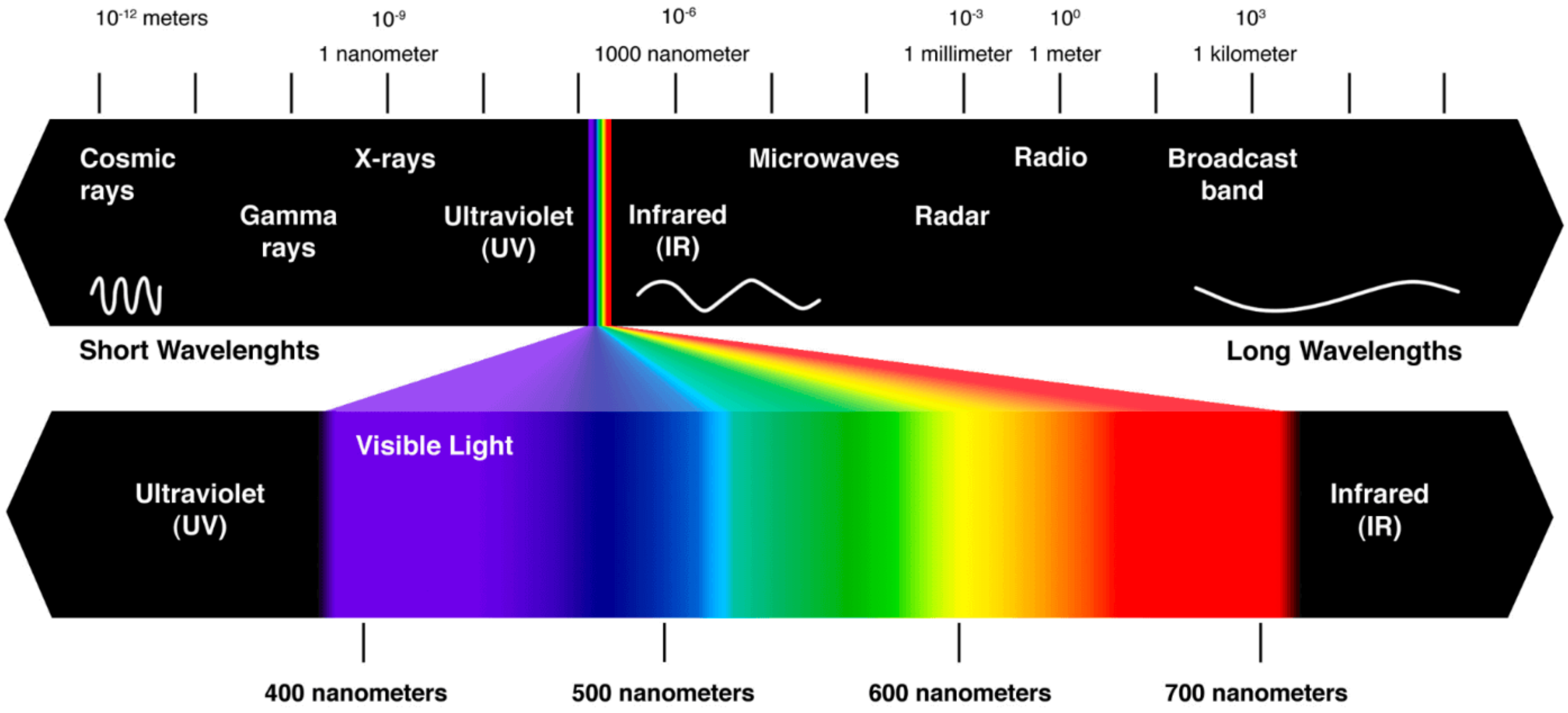


PHOTRESITORS & IMAGE SENSORS



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

LIGHT SPECTRUM



NEAR RED LIGHT

UV LIGHT

LIGHT PIXELATION

image is made of 113 wavelets

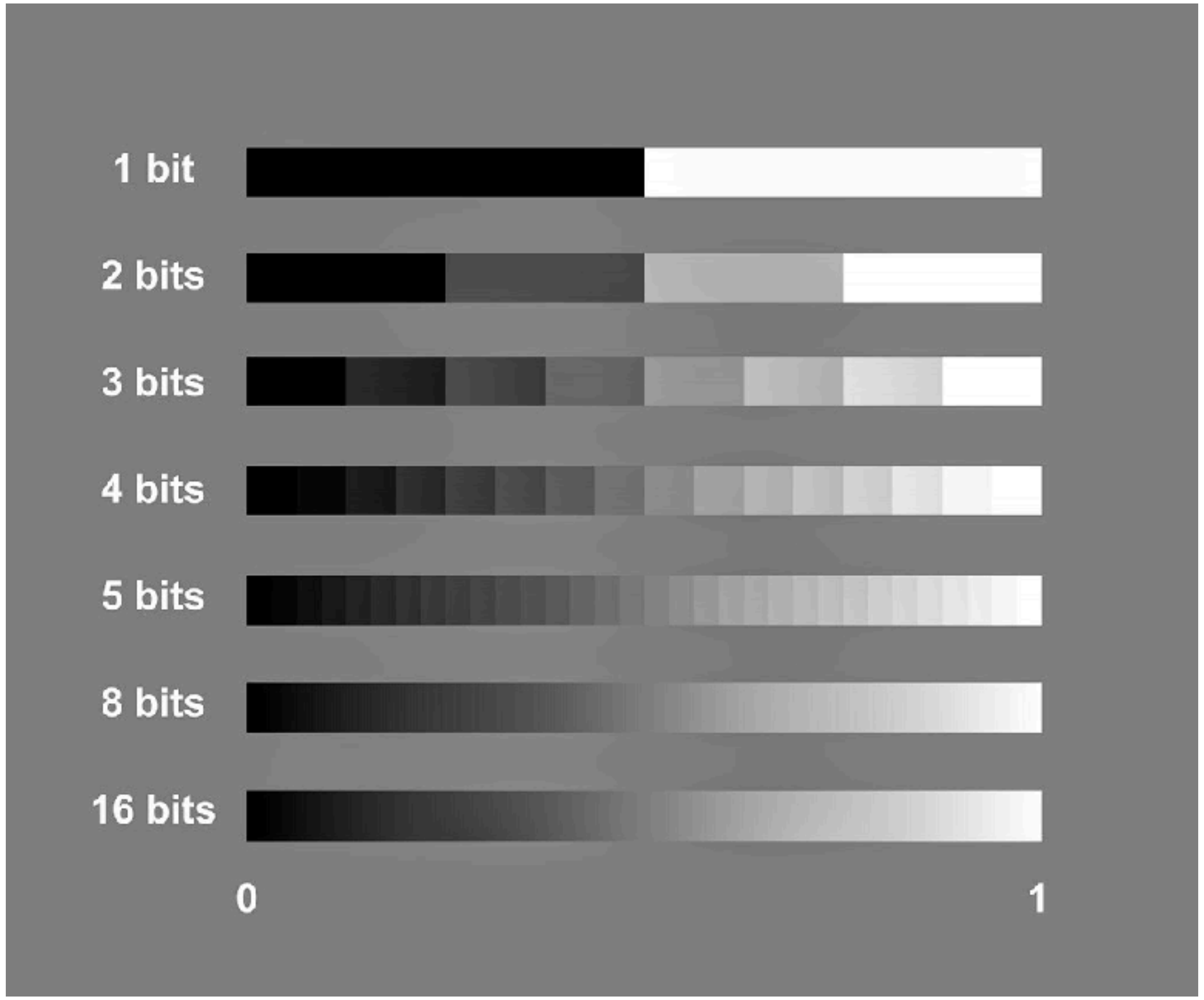


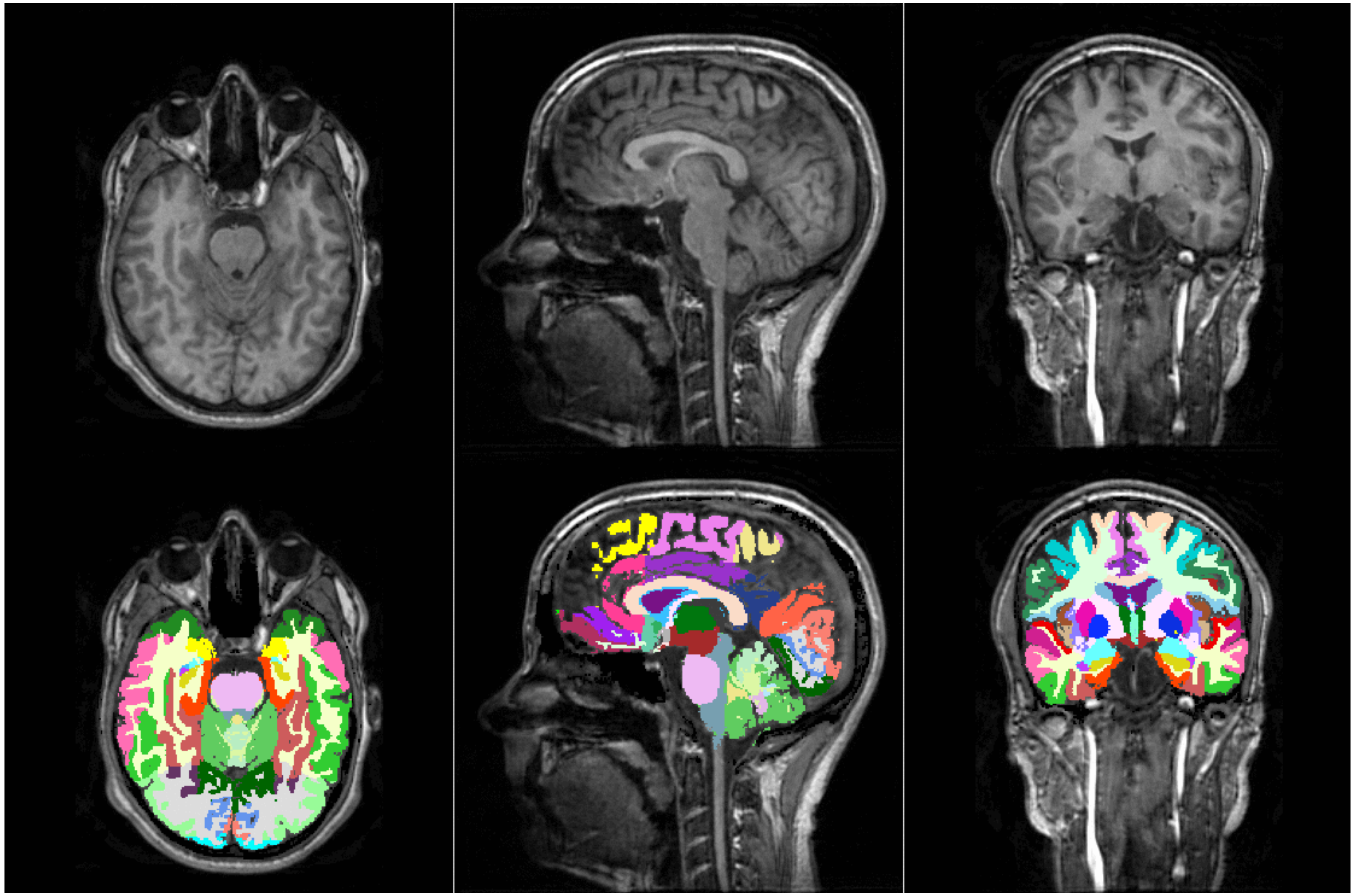
Image source

INTERACTION DESIGN

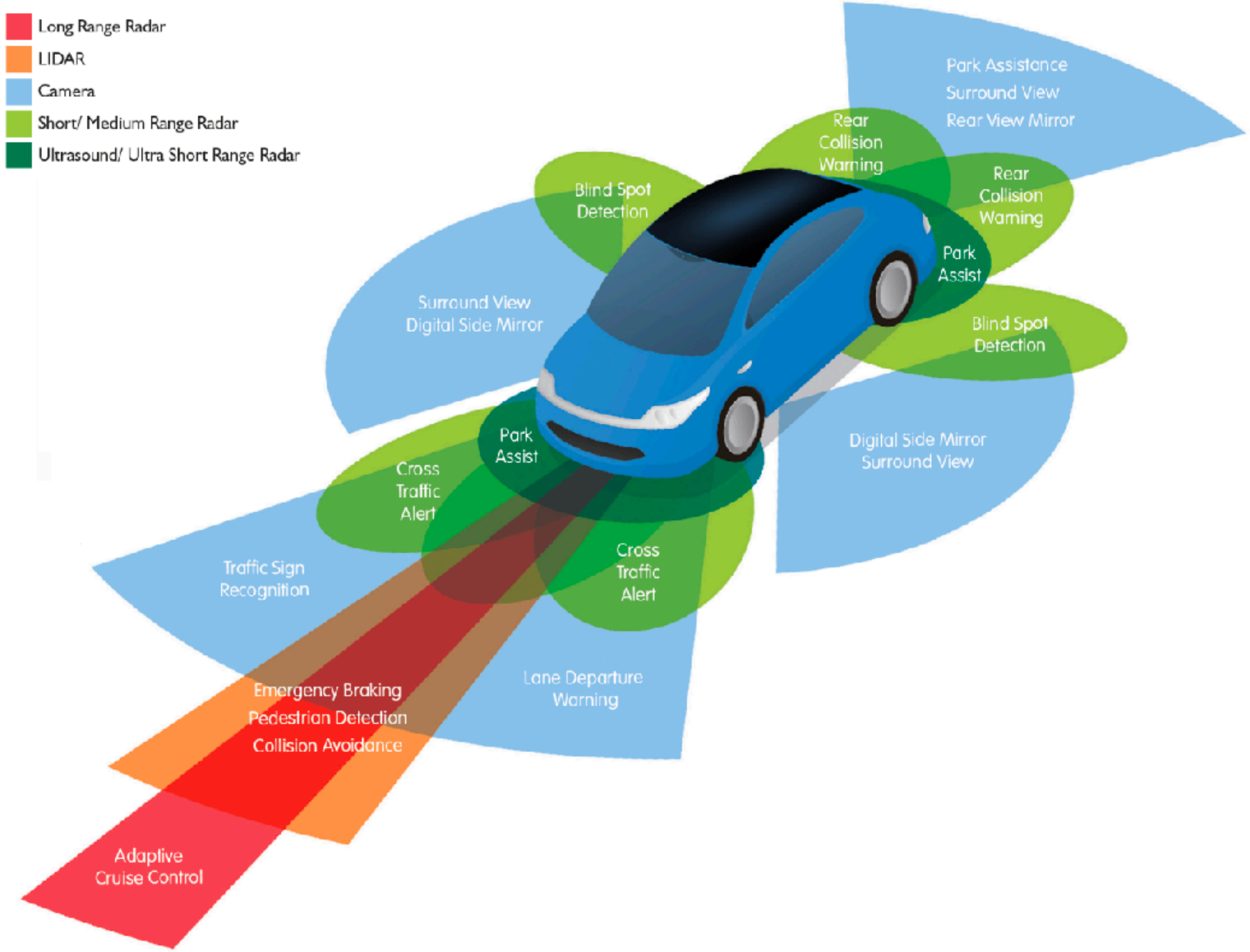
CV APPLICATIONS

Bits & Atoms IV

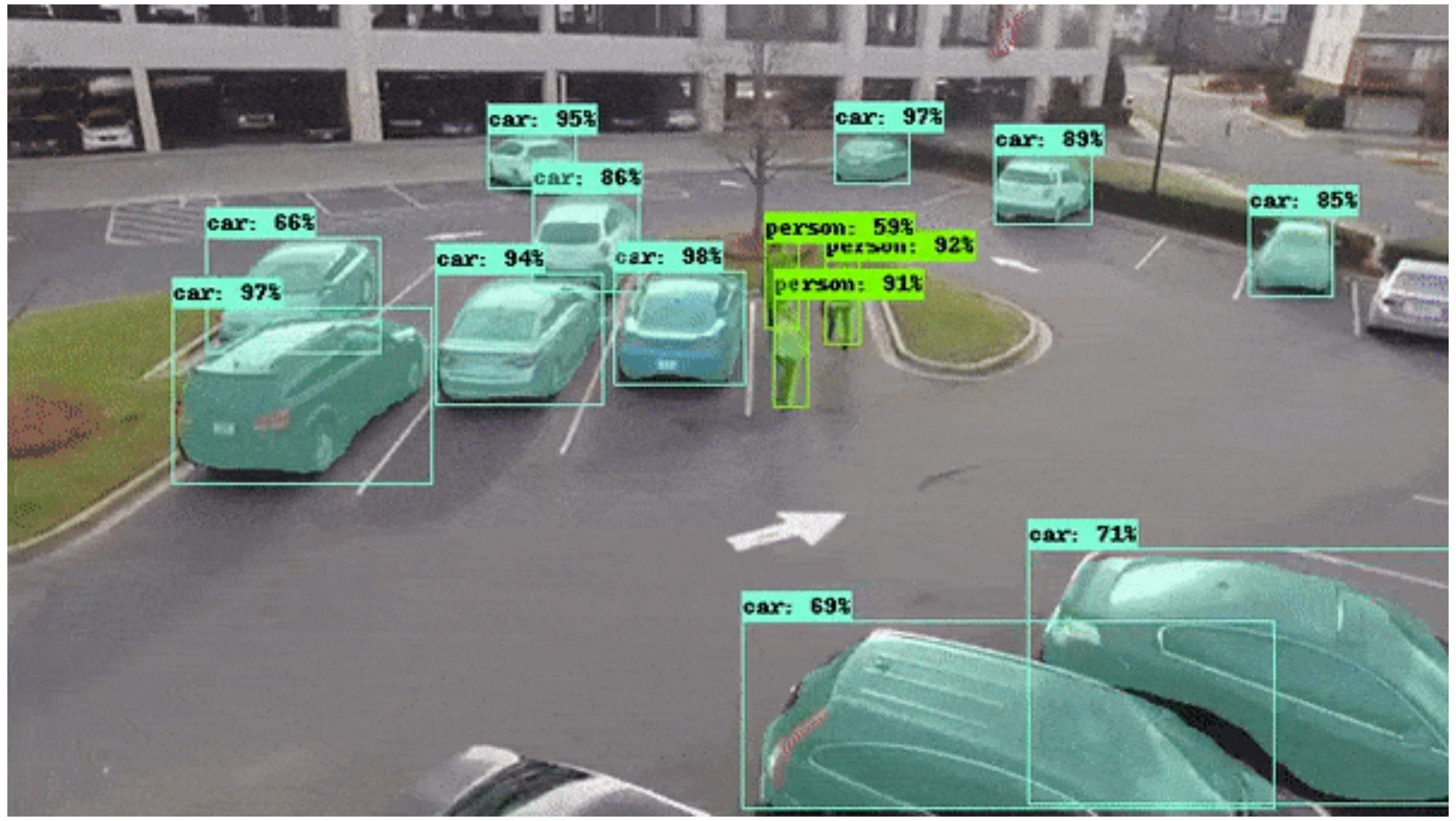
SCIENCE

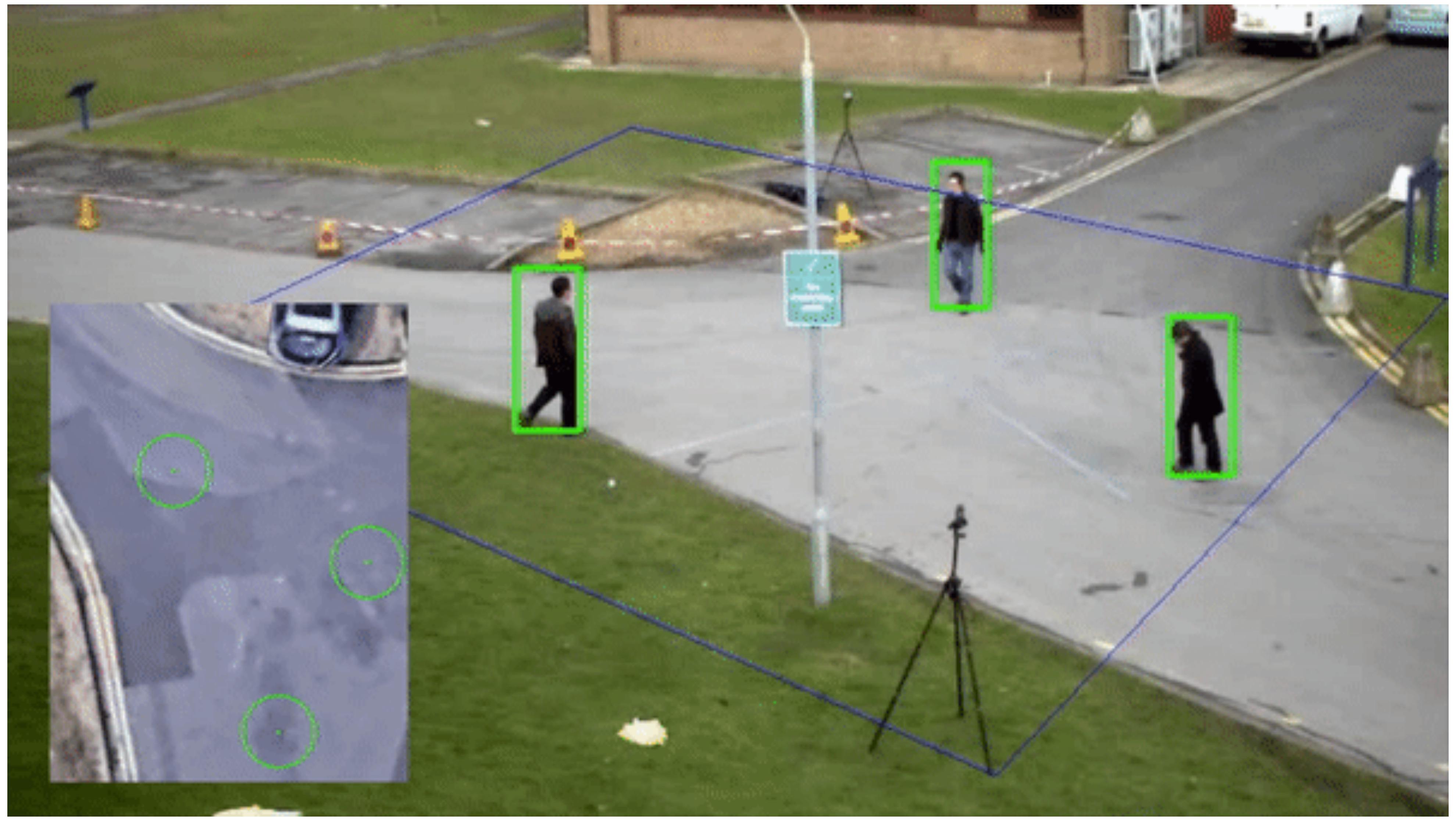


ADVANCED DRIVER ASSISTANCE

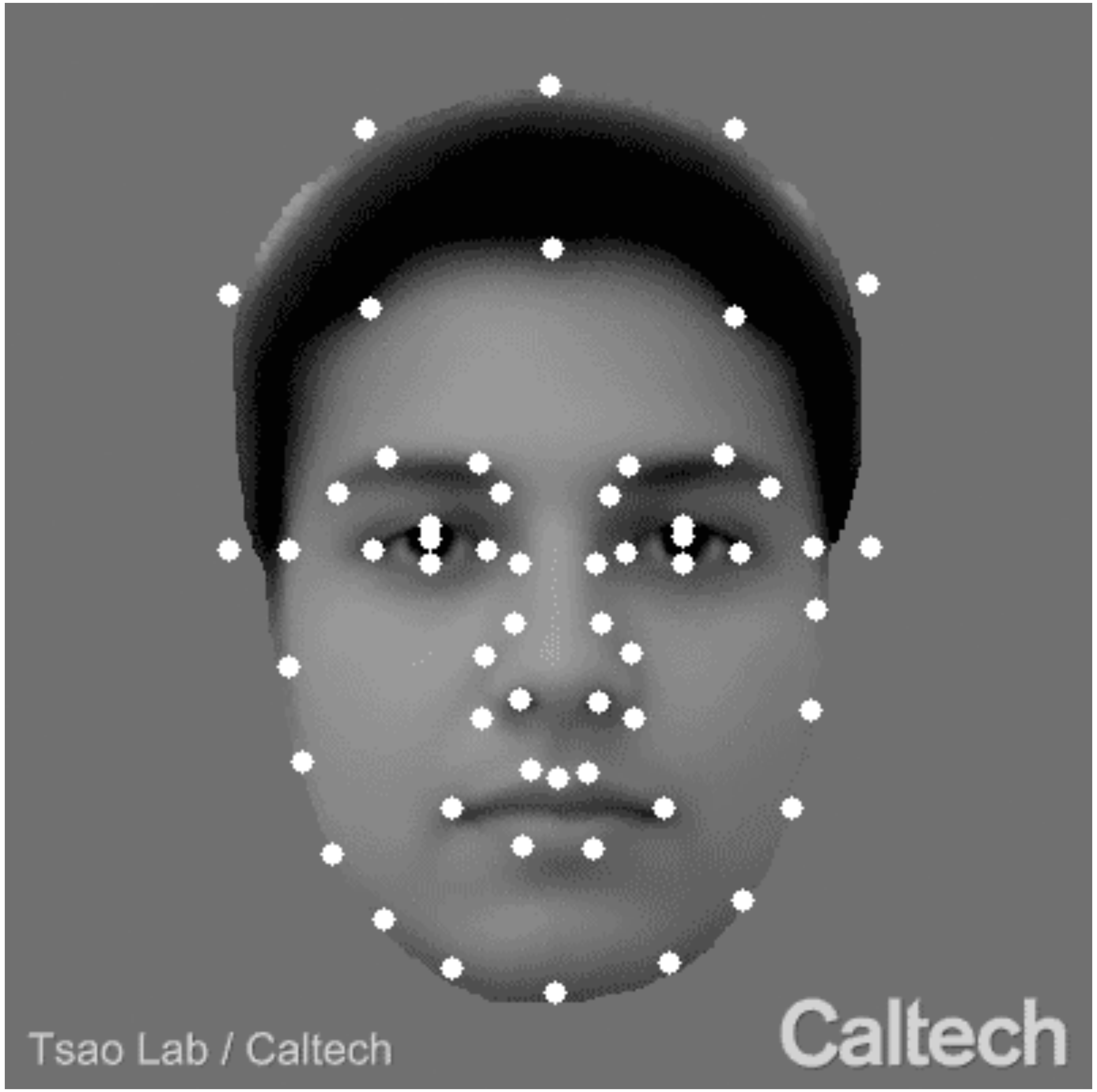


SECURITY/BEHAVIOUR RECOGNITION





FACIAL RECOGNITION



FACIAL RECOGNITION

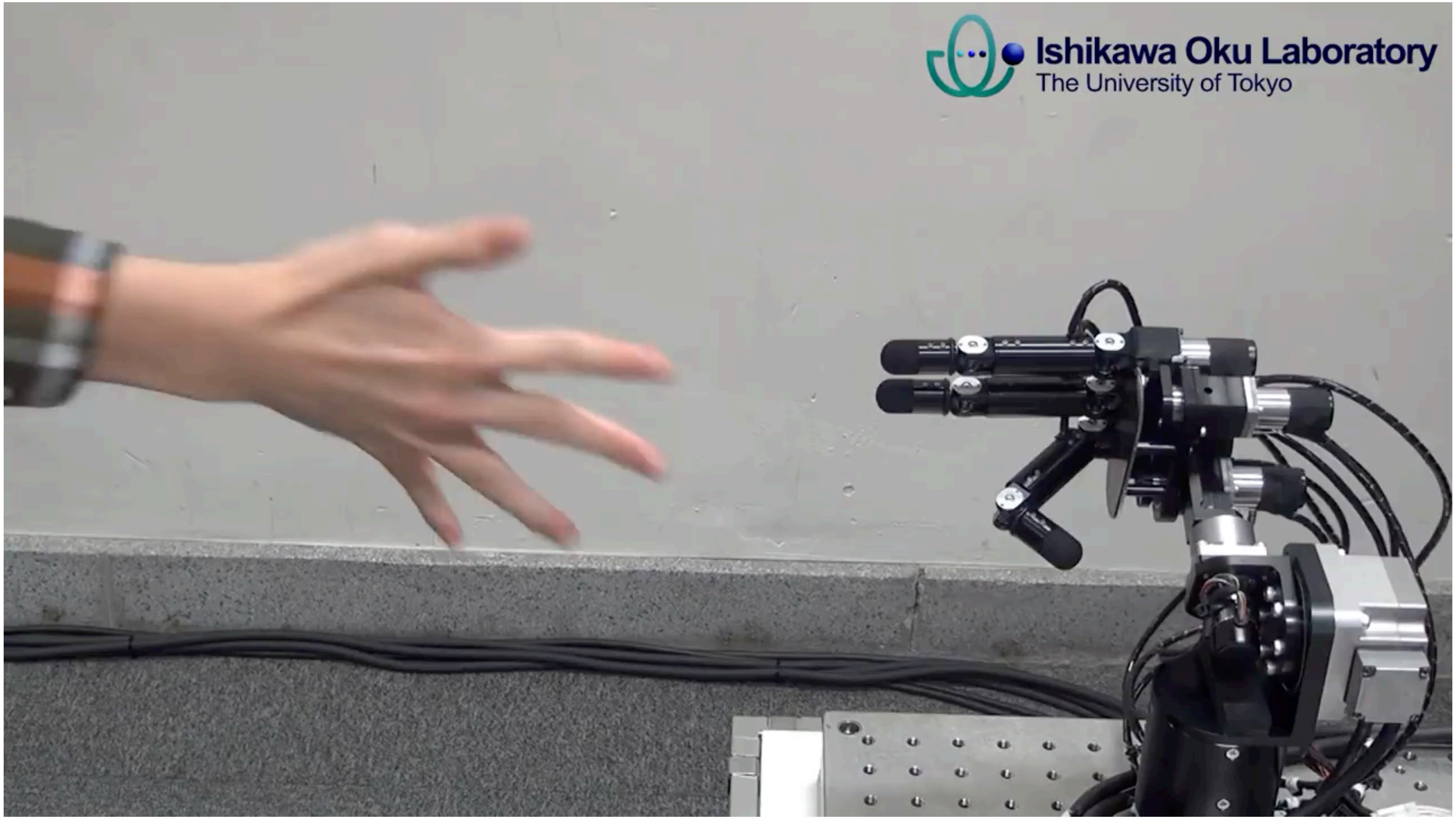
- **Biometric vulnerabilities** – those that occur in the algorithm of face detection, signature generation and liveness verification

- **Environmental vulnerabilities** – those that result from running the algorithm on the user’s end device

- **API vulnerabilities** – those that result from the modification of communication between the user and the identity verification server.

https://www.securing.pl/en/attacking-the-face-recognition-authentication-how-easy-is-to-fool-it/

ROBOTICS



PHOTOGRAMMETRY

CRITICAL DESIGN

ART & DESIGN

ART & DESIGN

HOW NORMAL AM I?

https://www.hownormalami.eu/

INTERACTION DESIGN

CV TOOLS

Bits & Atoms IV

COMPUTER VISION

Many popular computer vision applications involve trying to recognise things in video or image.

COMPUTER VISION

Object Classification: Broad category of the objects

Object Identification: Type of the objects

Object Verification: Are the objects visible at all?

Object Detection: Position of the objects

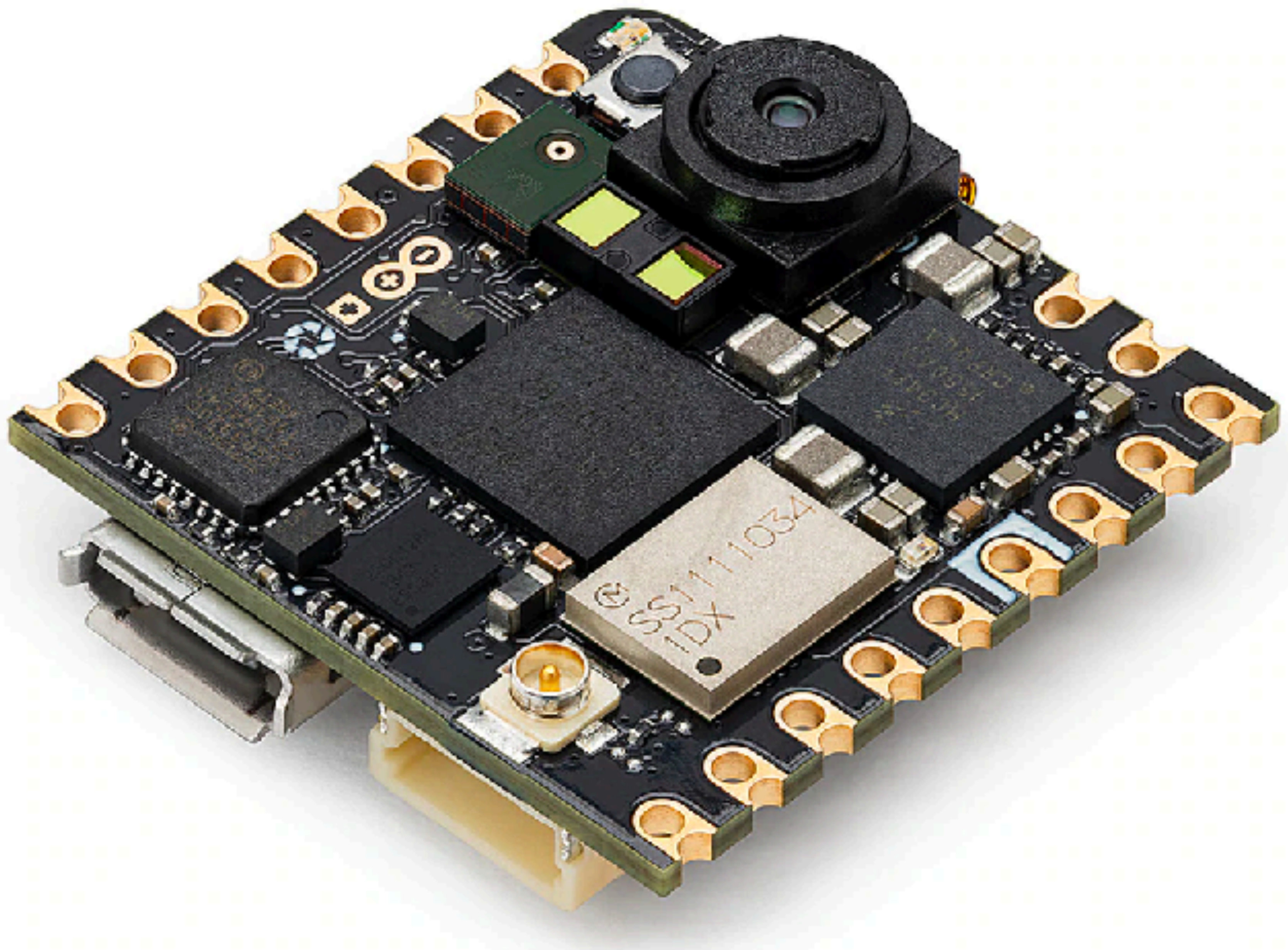
Object Landmark Detection: Key points for the objects

Object Segmentation: Pixels belonging to the objects

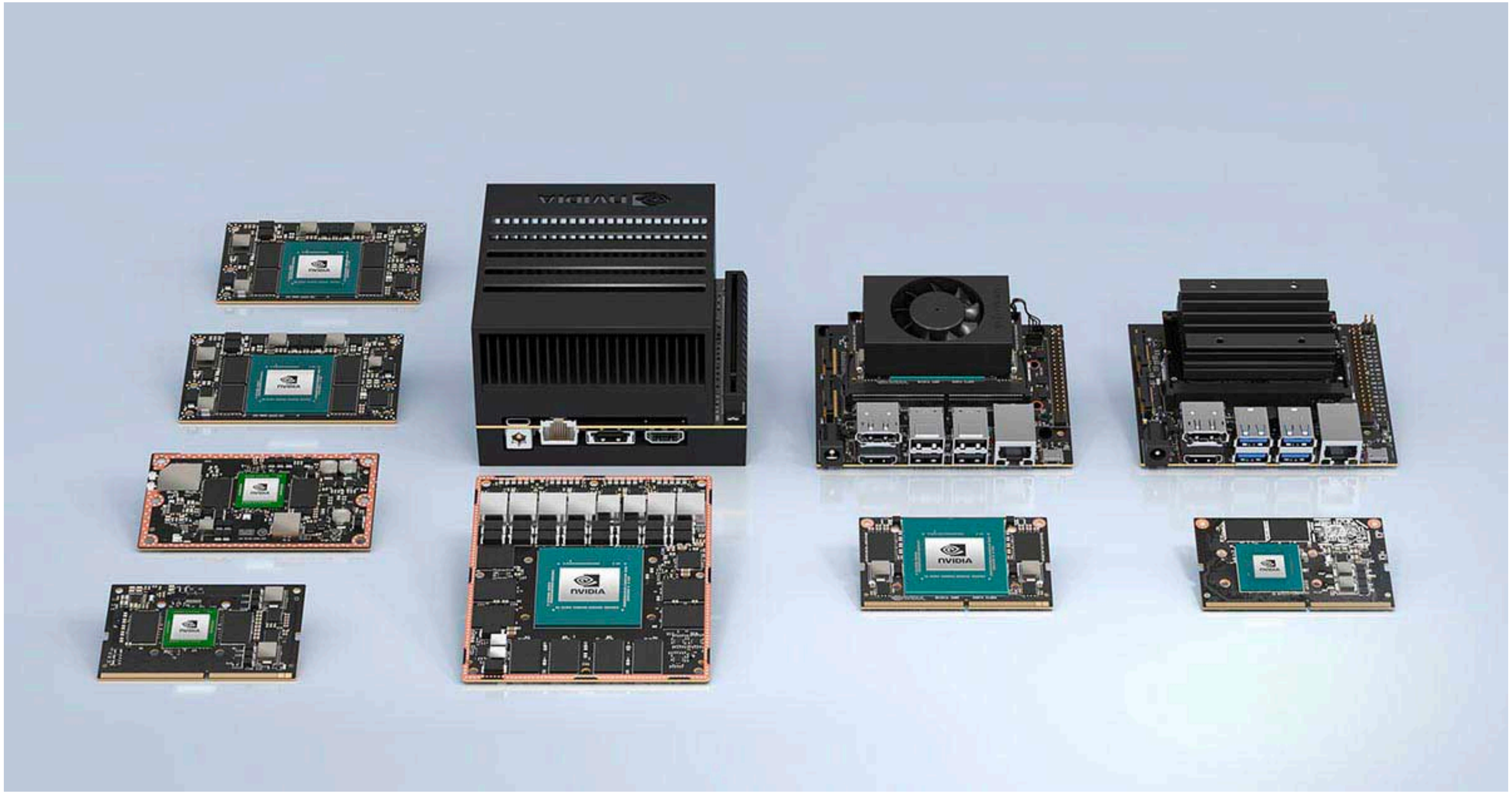
Object Recognition: Are the given objects in the image?

REAL SENSE

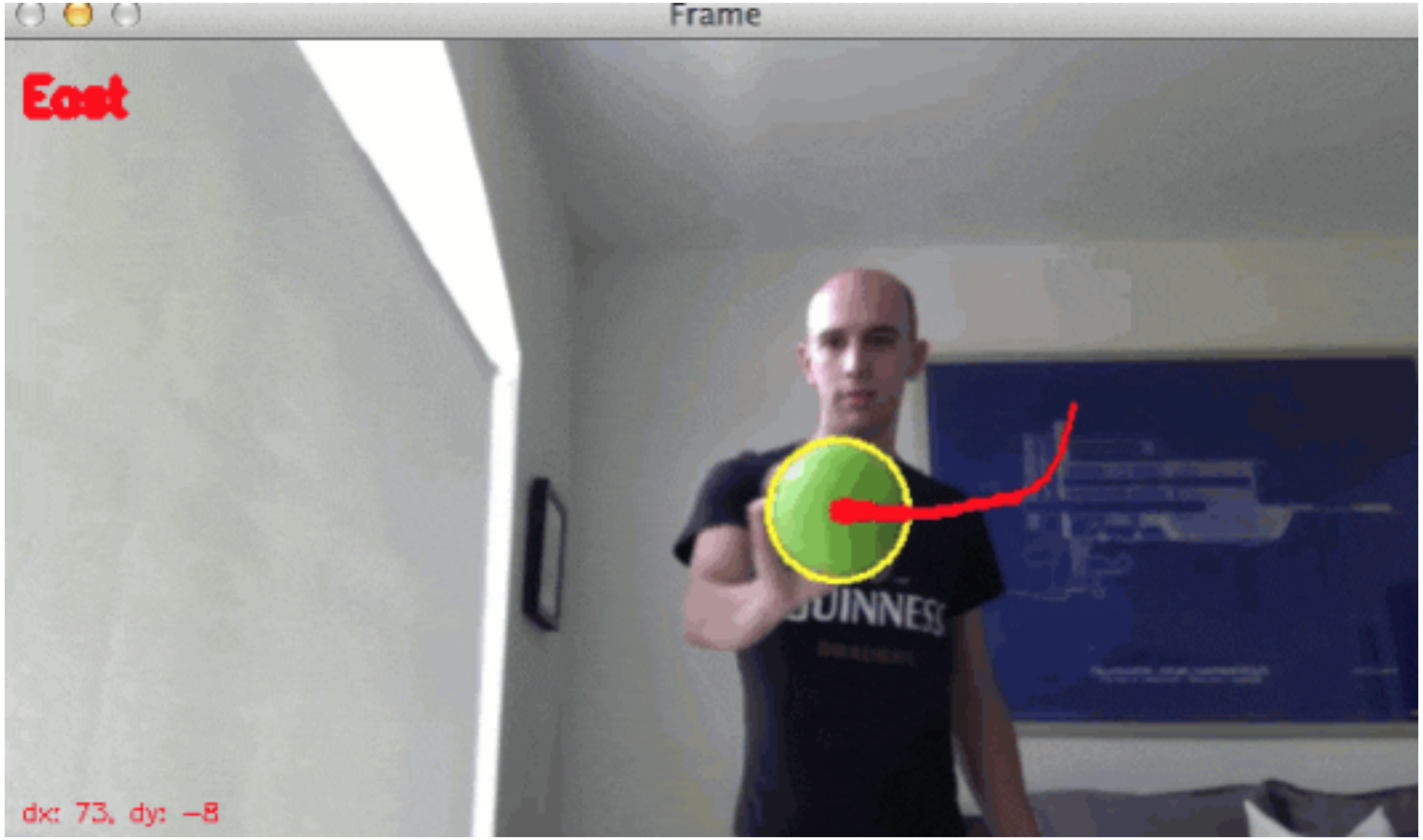
OPEN MV /NICLA VISION



NVIDIA JETSON



OPEN CV

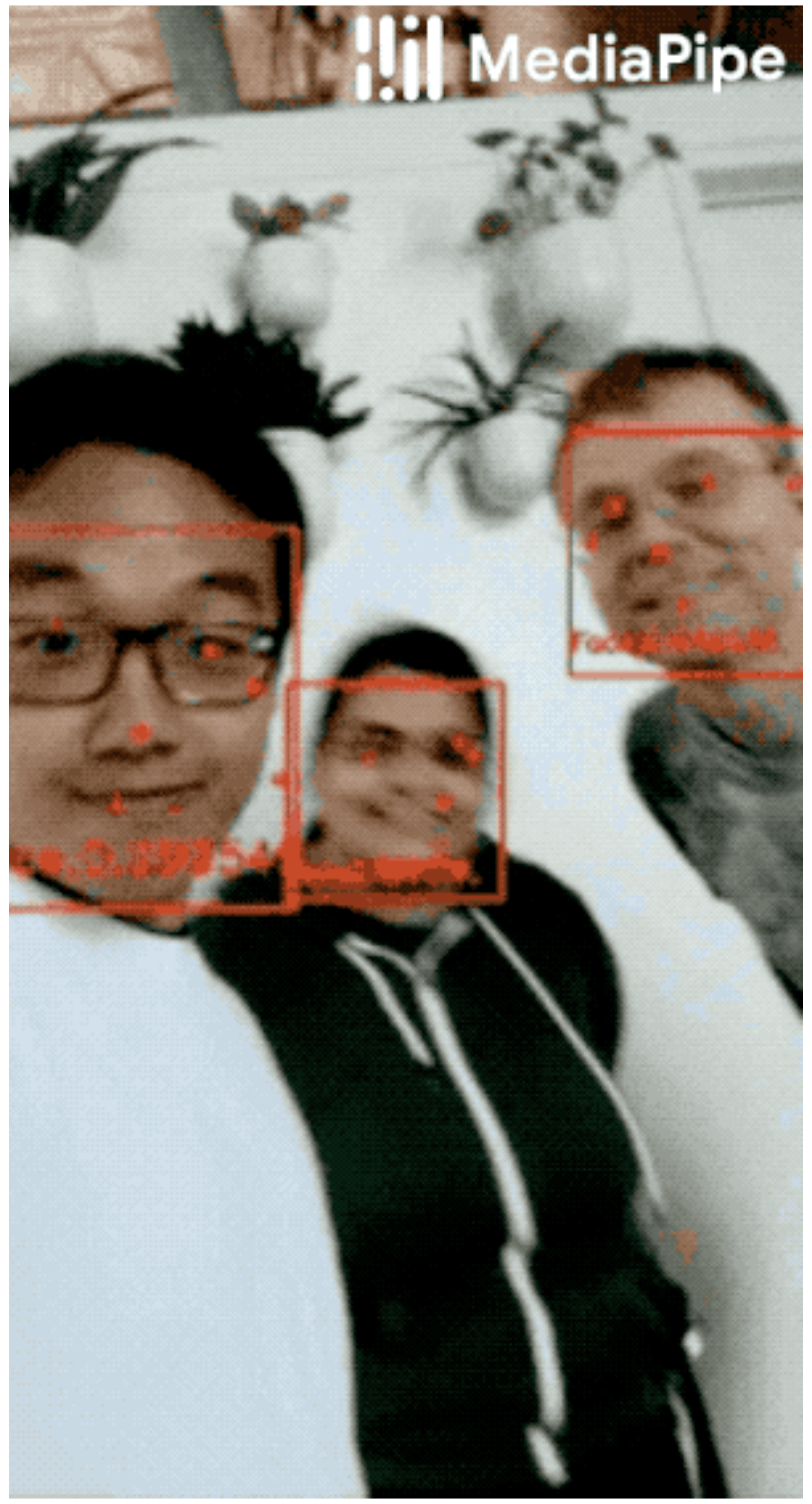


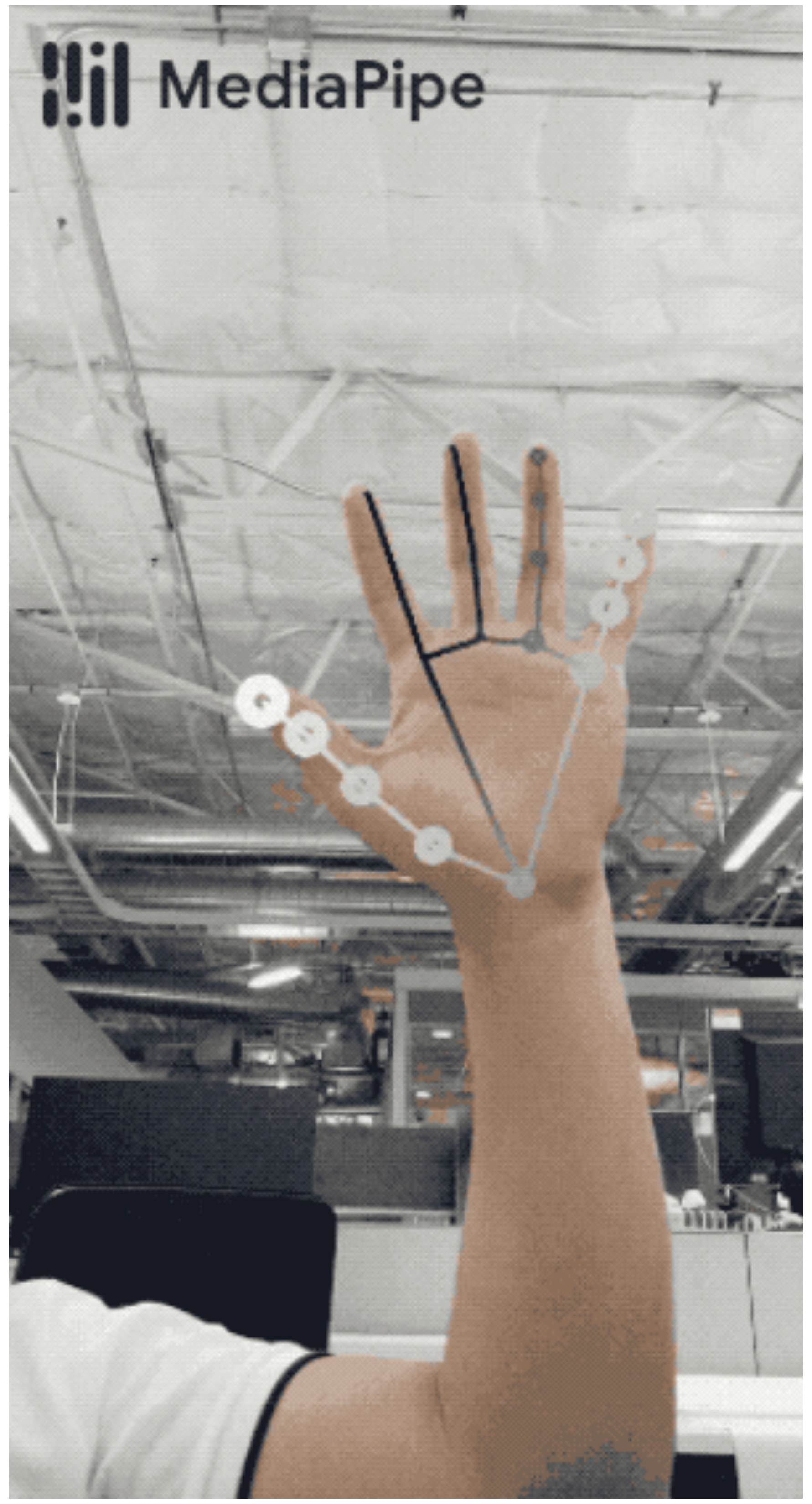
https://opencv.org/ https://github.com/atduskgreg/opencv-processing https://github.com/orgicus/p5.js-cv

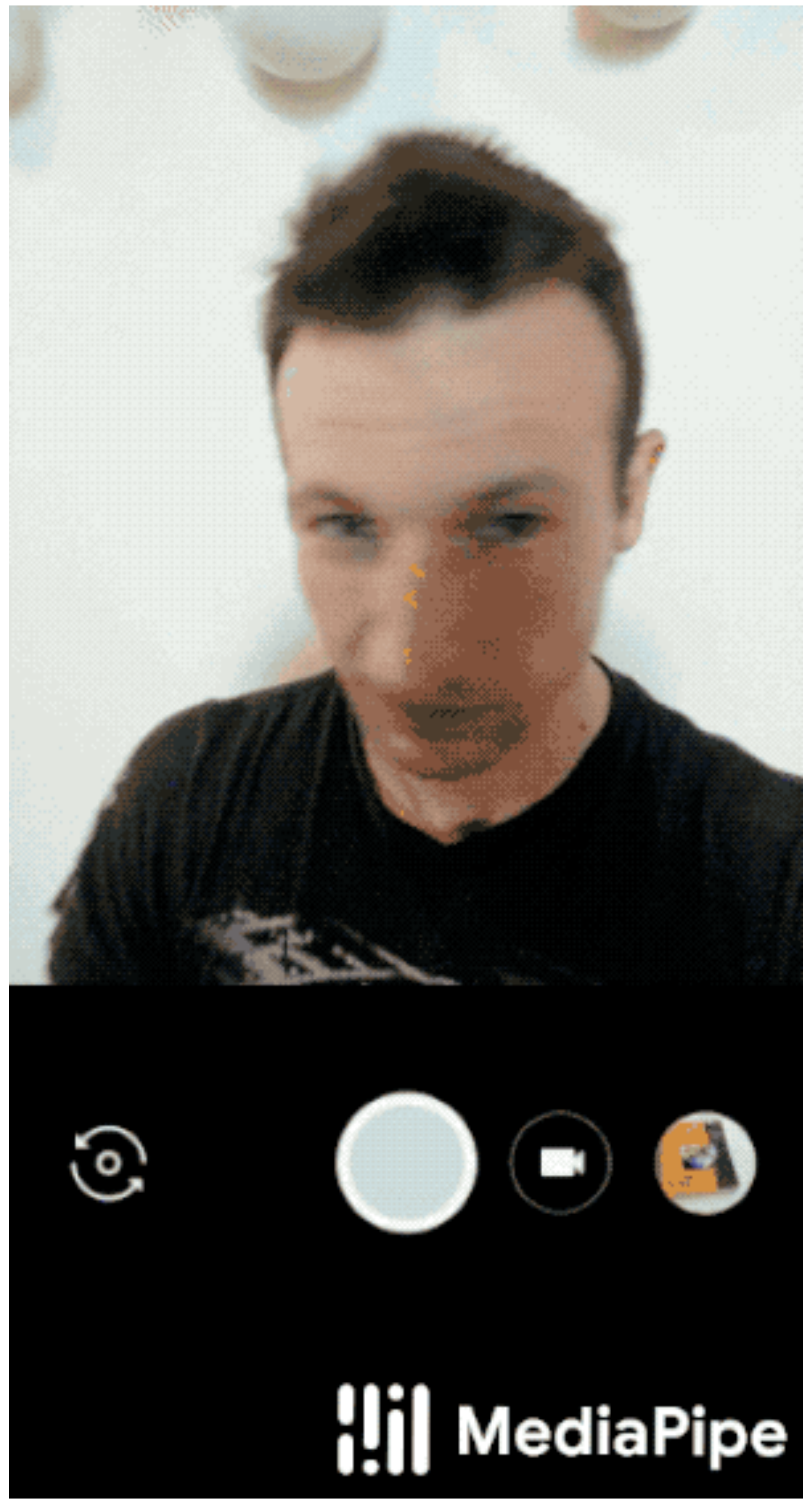
DETECTRON

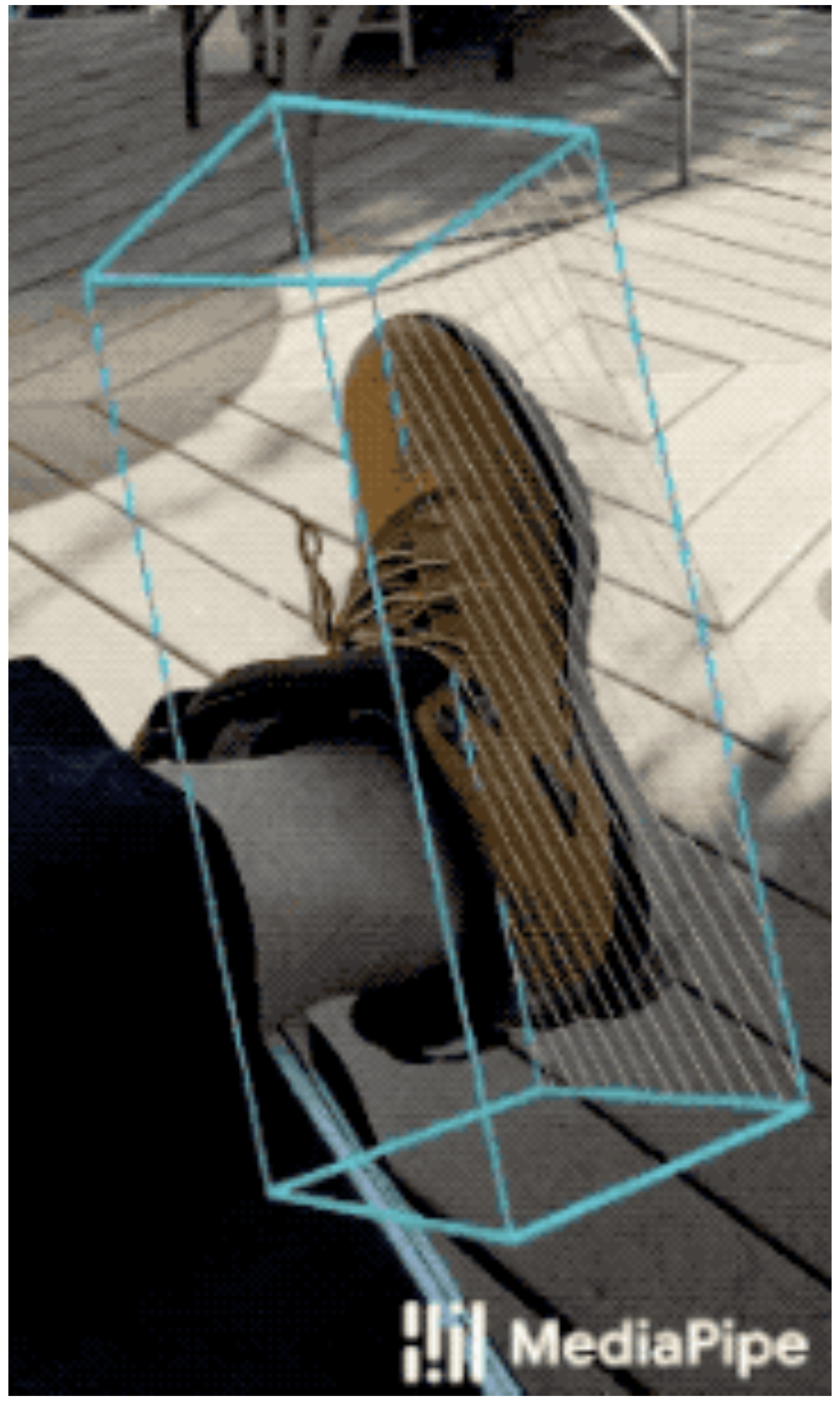

sky
tree
person 100%
person 99%
backpack 90%
person 91%
person 67%
person 69%
person 56%
person 93%
person 99%
person 74%
person 97%
person 99%
person 69%
bicycle 91%
bicycle 62%
person 100%
bicycle 96%
bottle 77%
bicycle 98%
road

MEDIAPIPE



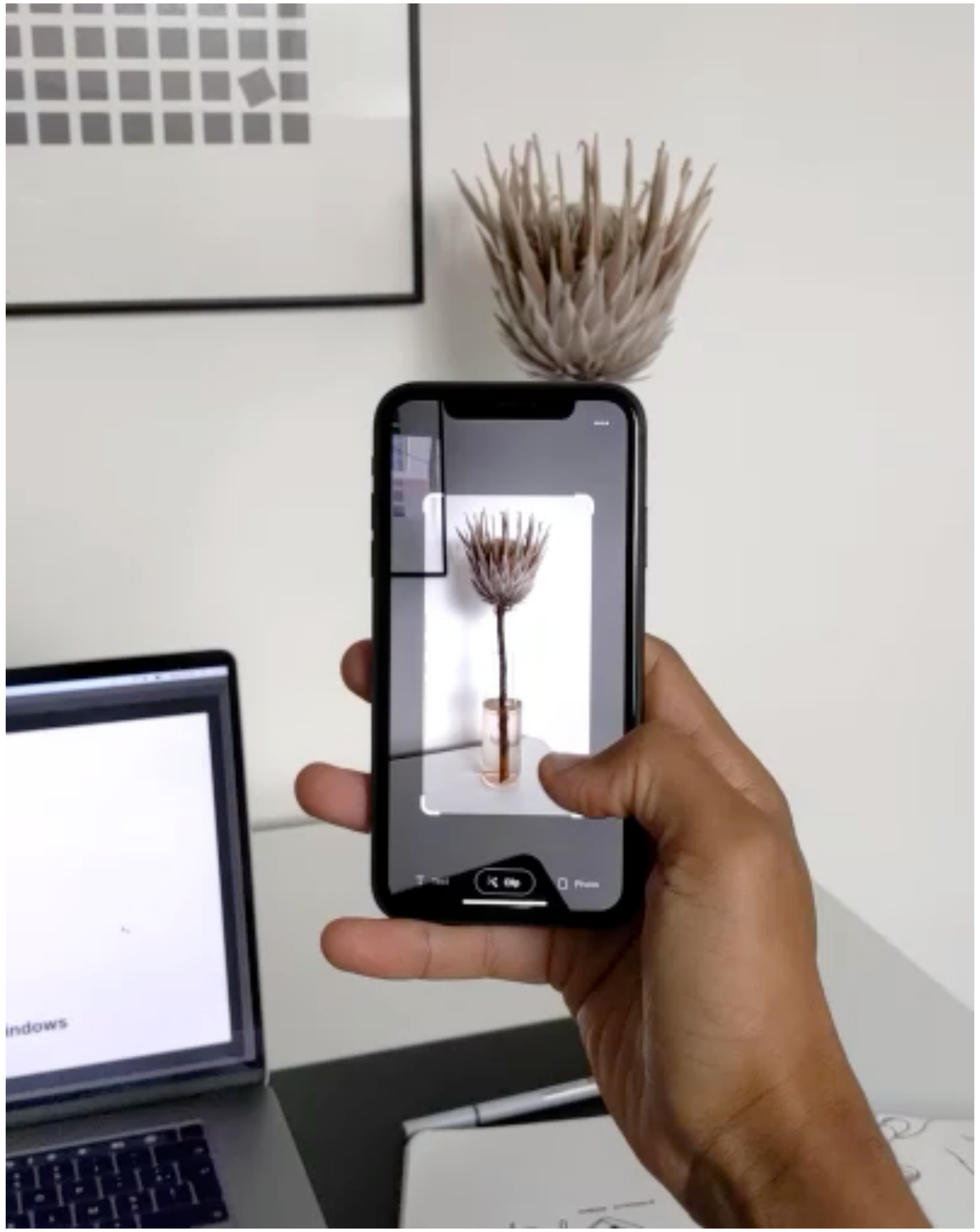




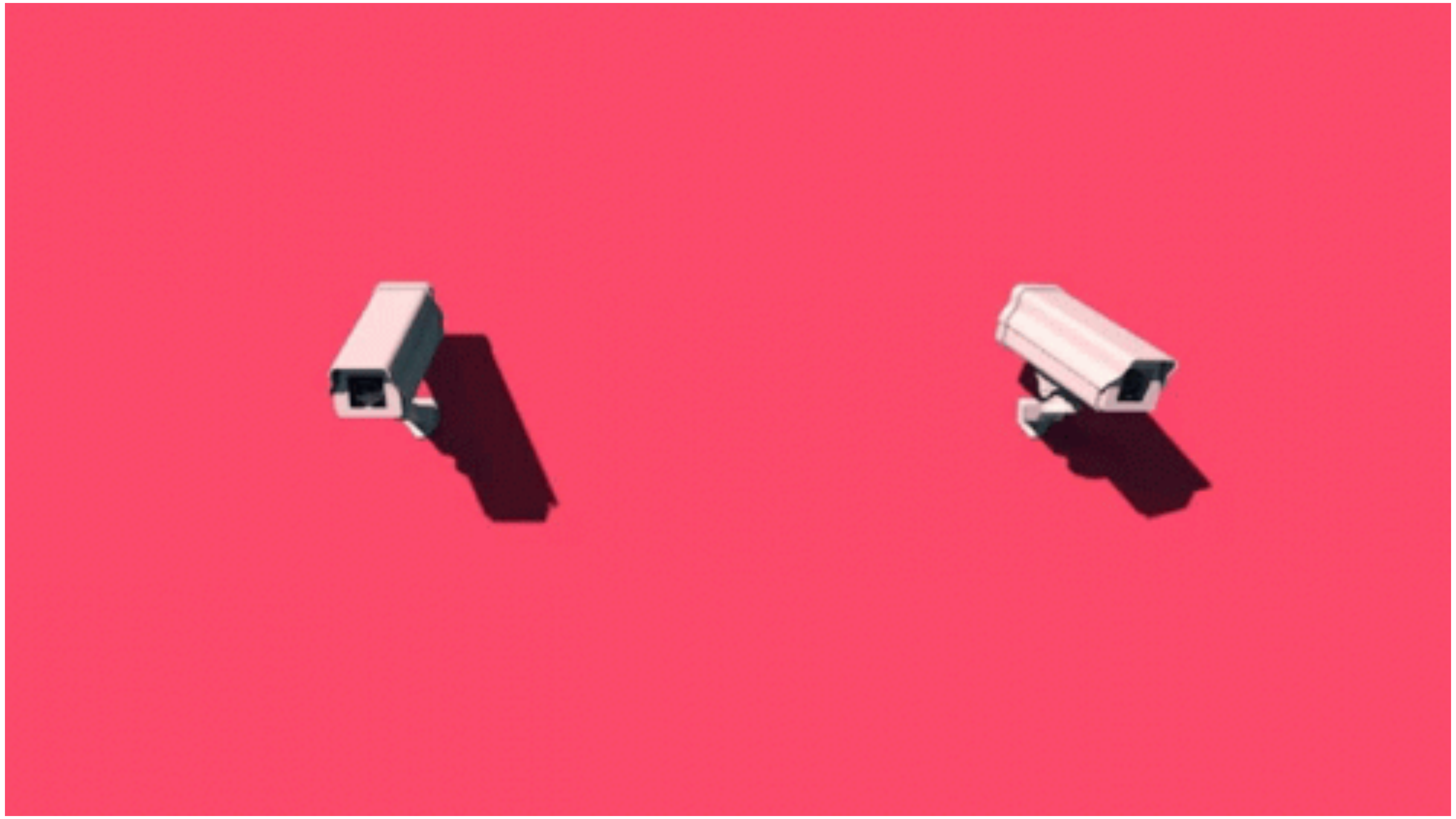


OPEN DATA CAM

CLIPDROP



TRACKING.JS

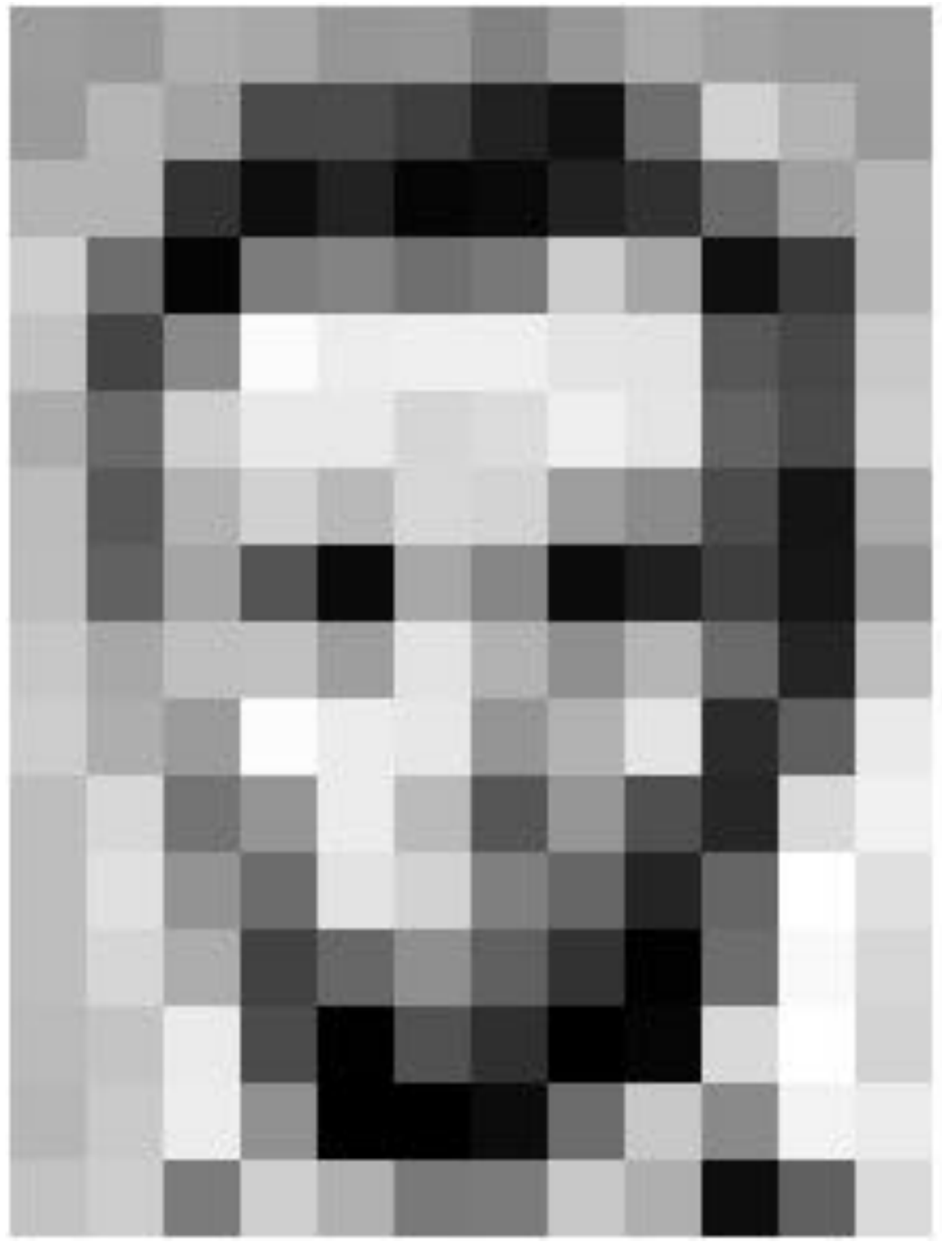


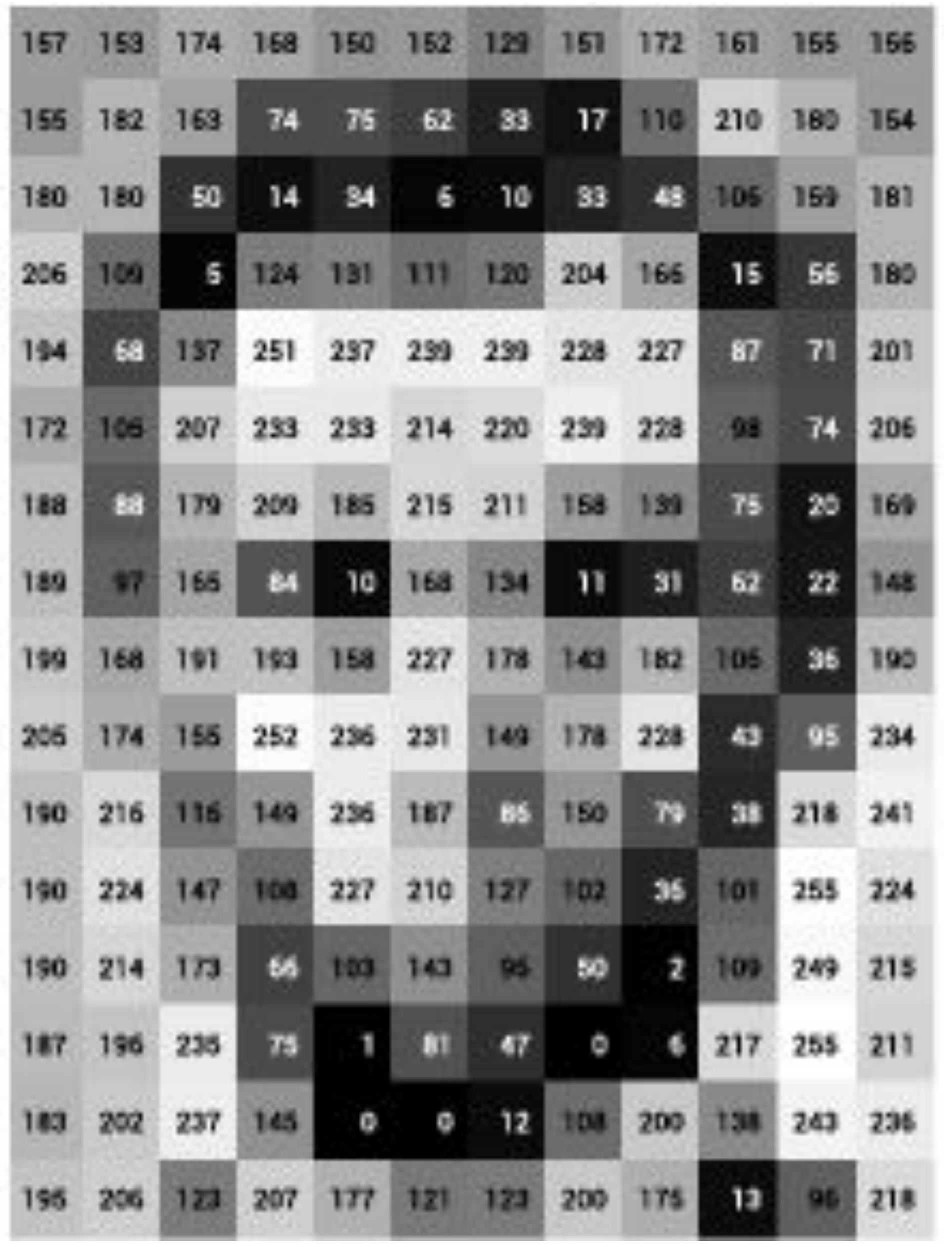
INTERACTION DESIGN

IMAGE PROCESSING IN P5.JS

Bits & Atoms IV

IMAGE



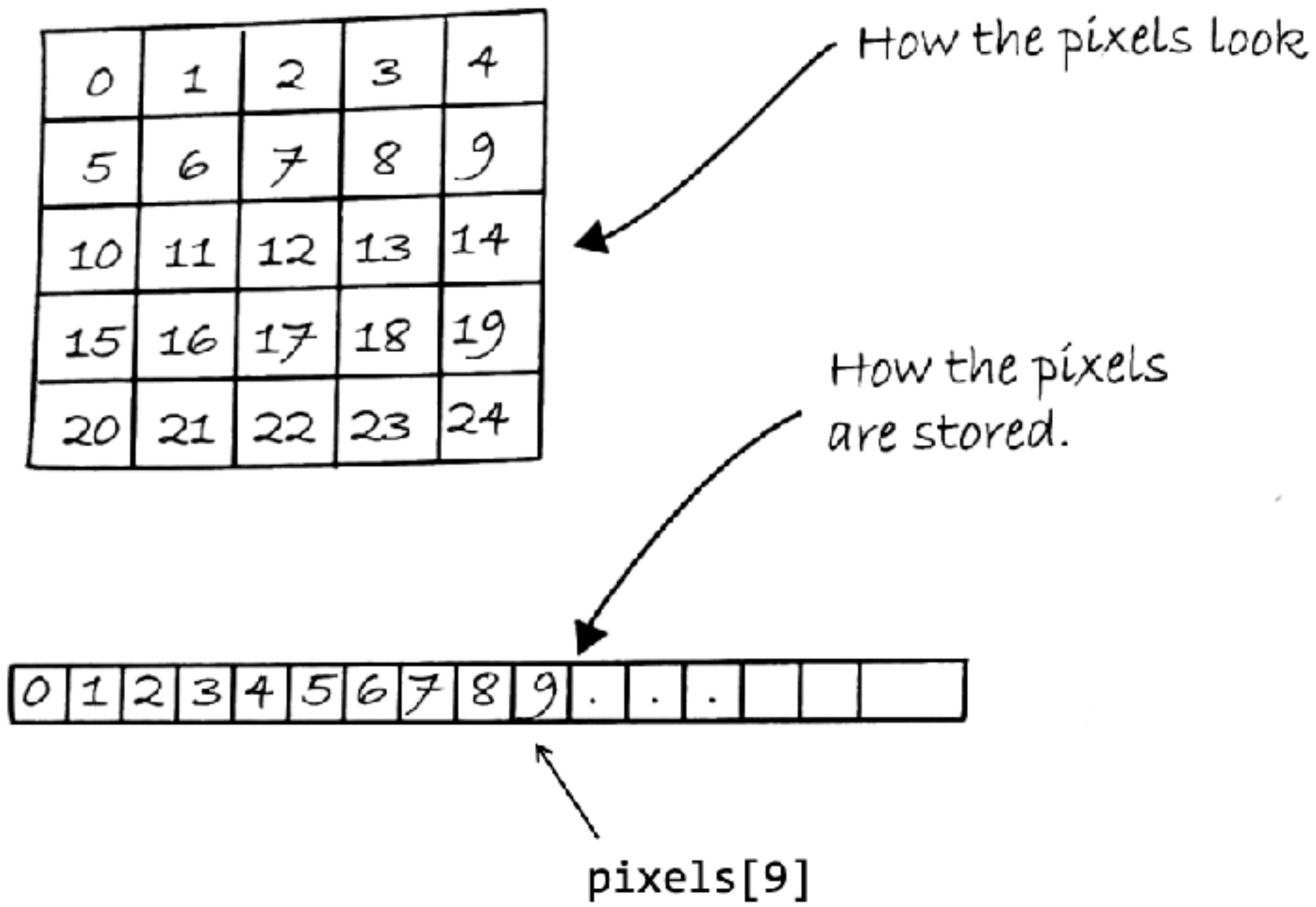


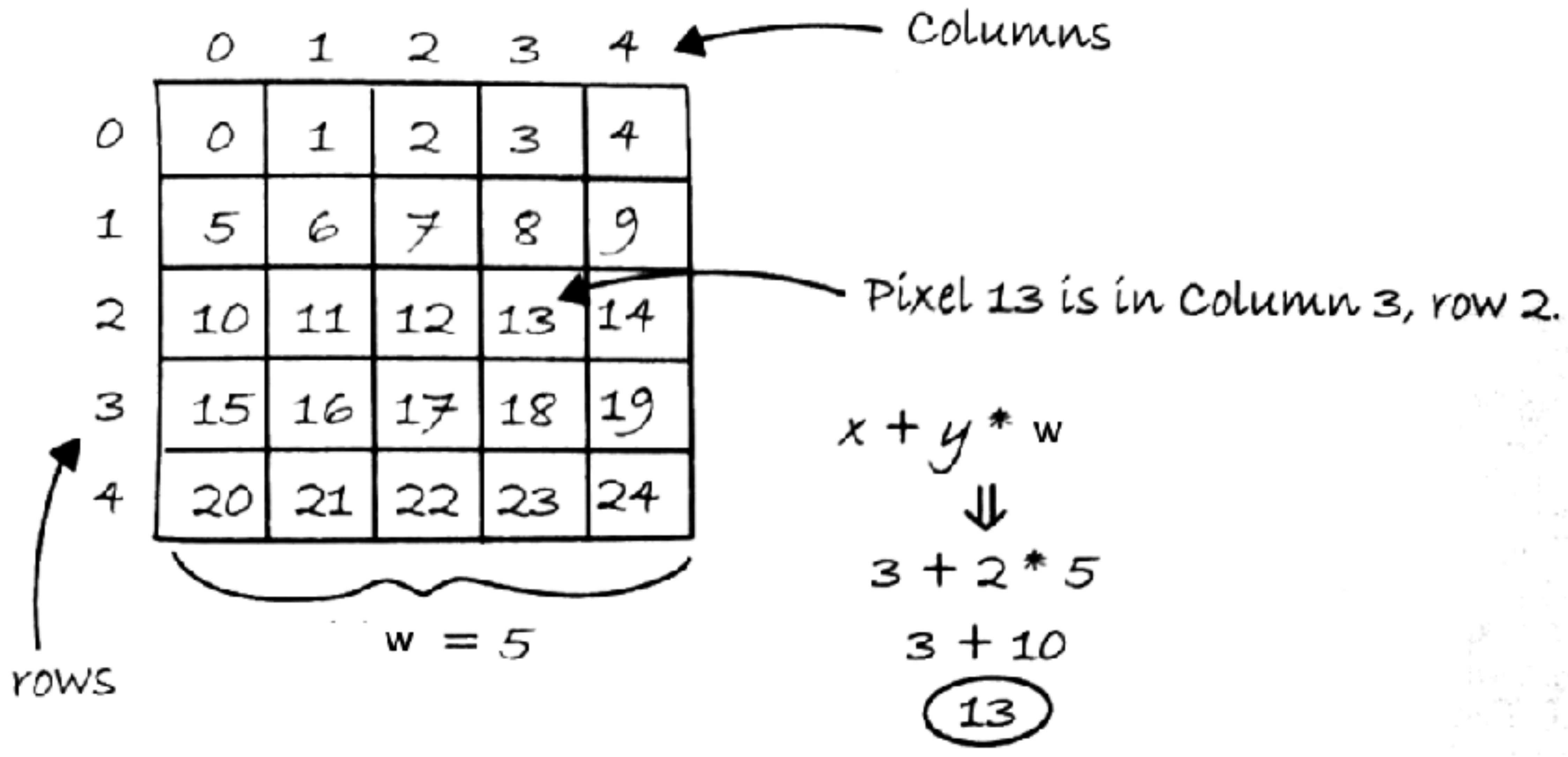
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

width * height = pixels

Image source

IMAGE






```
let index = x + y * width
```

IMAGE

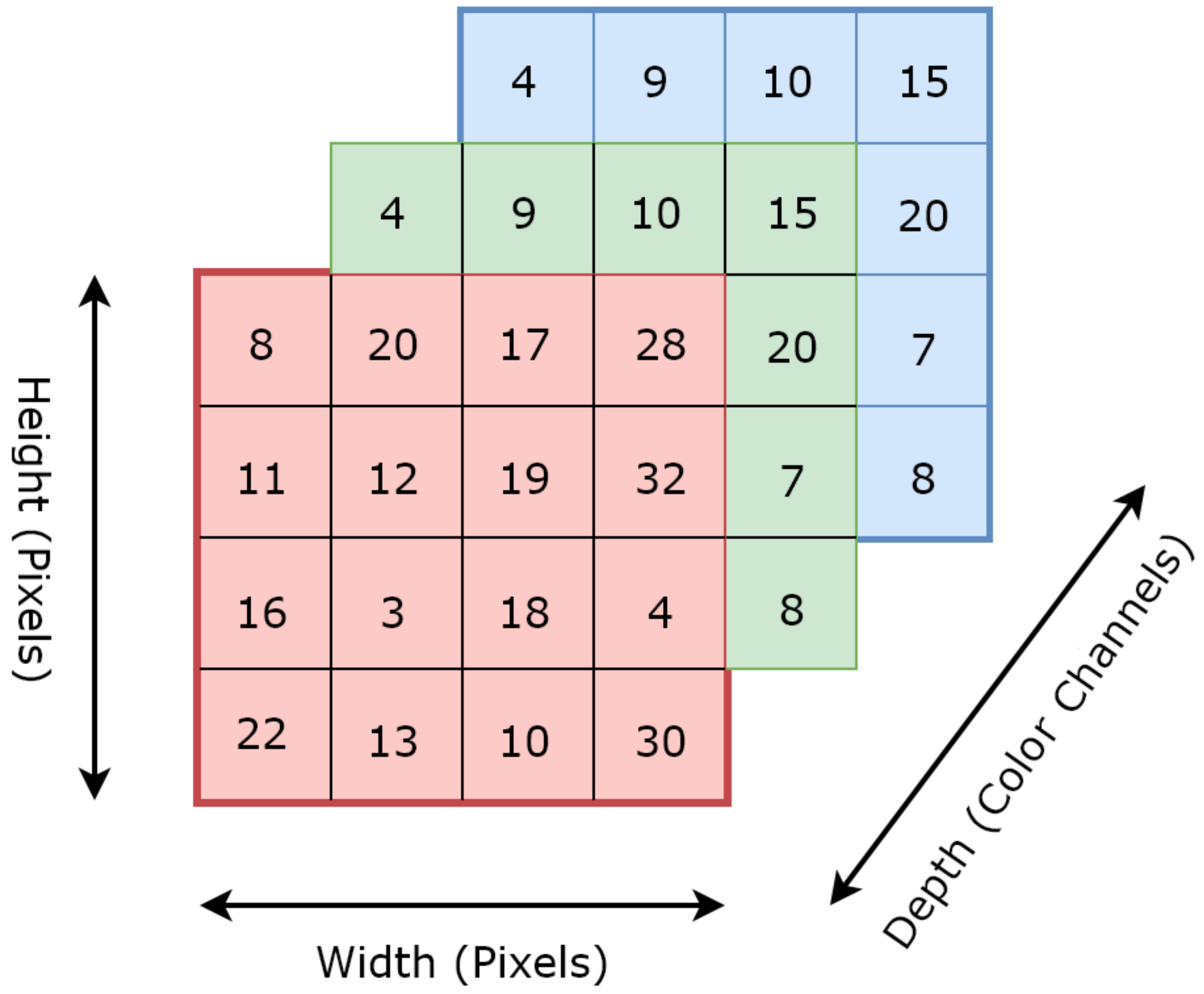


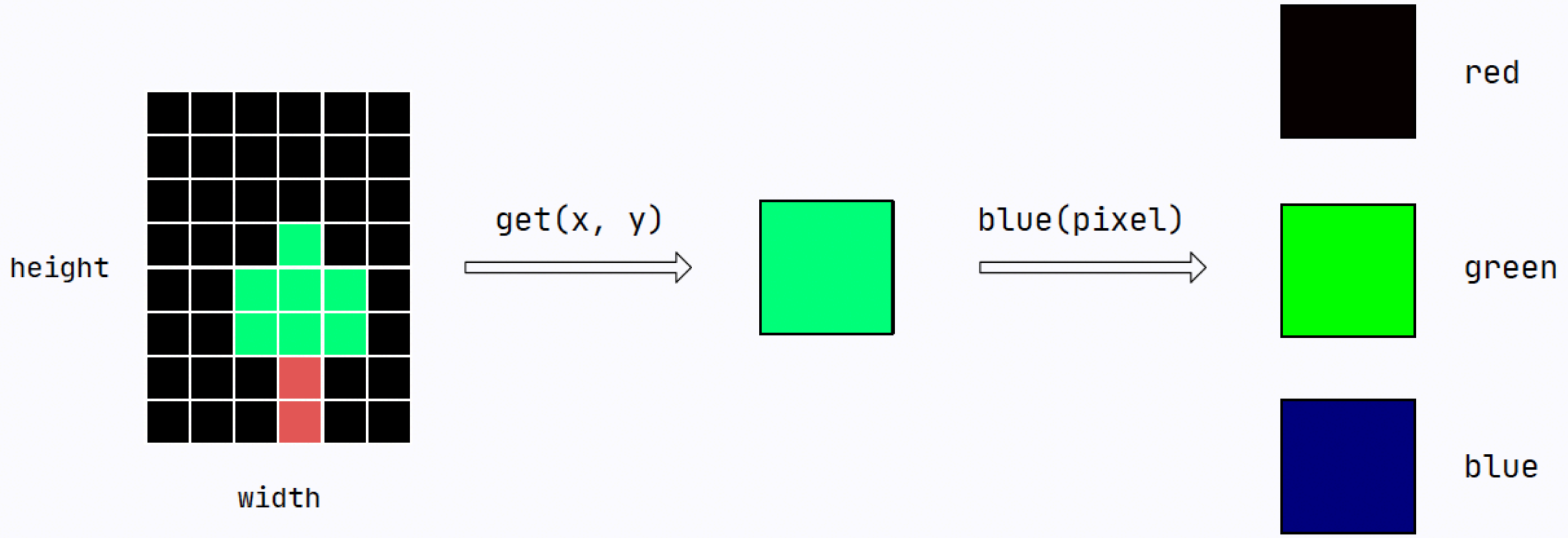
Image source

IMAGE

The memory used to store images is called the video buffer or frame buffer.

Direct access to the screen’s frame buffer is pretty unusual on modern computers, and high level libraries like p5.js don’t (can’t) provide it.

IMAGE



IMAGE

```
Let pixel = img.get(x, y);

Let r = red(pixel);
Let g = green(pixel);
Let b = blue(pixel);
Let h = hue(pixel);

Let s = saturation(pixel);
Let b = brightness(pixel);

img.set(x, y, color(0))
```

IMAGE

```
// getRGBA()
// find the RGBA values of the pixel at `x`, `y` in the pixel array of `img`
// unlike get() this functions only supports getting a single pixel
// it also doesn't do any bounds checking or other checks


function getRGBA(img, x, y) {
  const i = (x + y * img.width) * 4;
  return [
    img.pixels[I],      //R
    img.pixels[i + 1],  //G
    img.pixels[i + 2],  //B
    img.pixels[i + 3],  //A
  ];
}
```

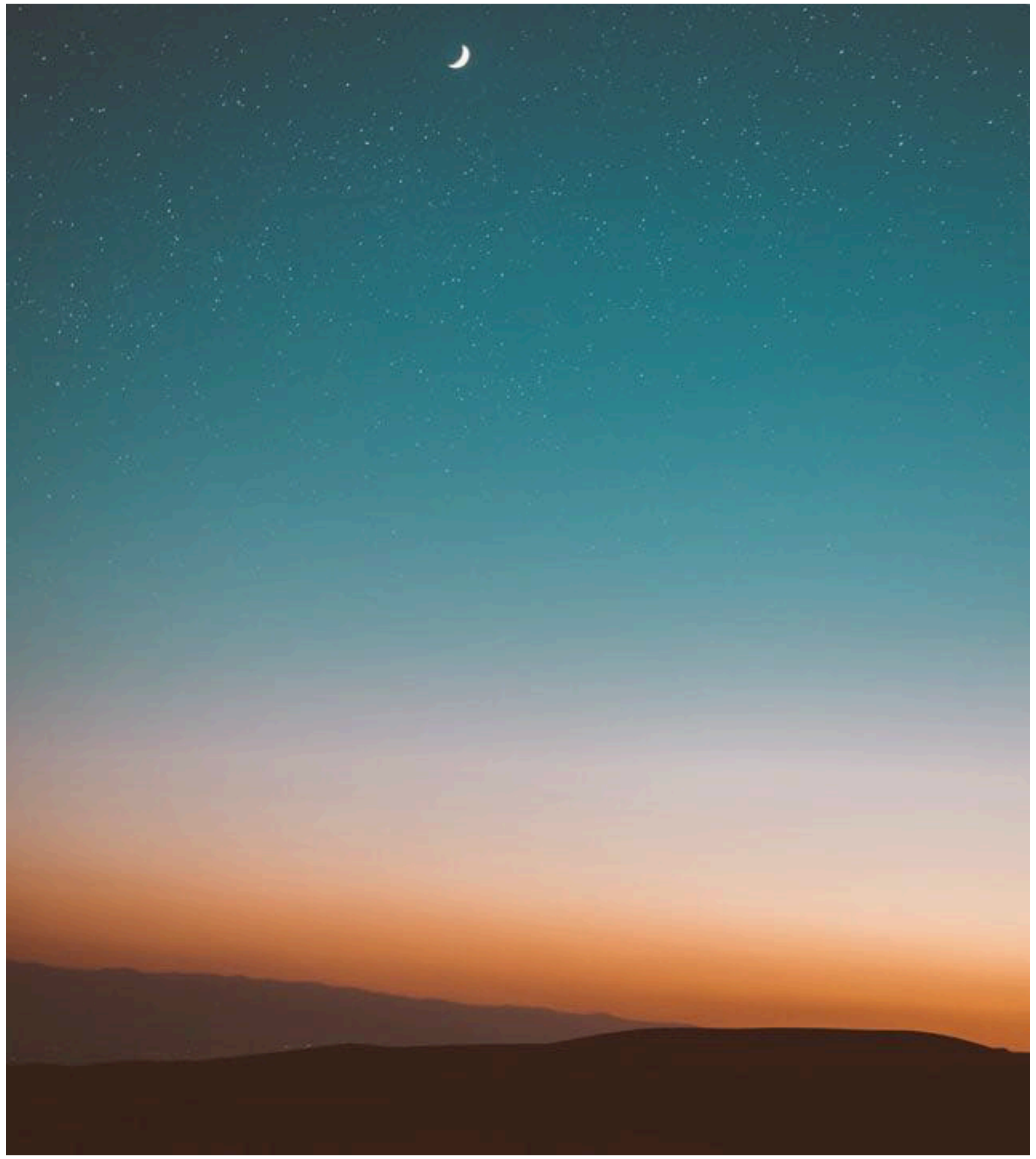
IMAGE

```
// setRGBA()
// set the RGBA values of the pixel at `x`, `y` in the pixel array of `img`
// unlike set() this functions only supports setting a single pixel

function setRGBA(img, x, y, c) {
  const i = (x + y * img.width) * 4;

  img.pixels[i]     = c[0];
  img.pixels[i + 1] = c[1];
  img.pixels[i + 2] = c[2];
  img.pixels[i + 3] = c[3];
}
```

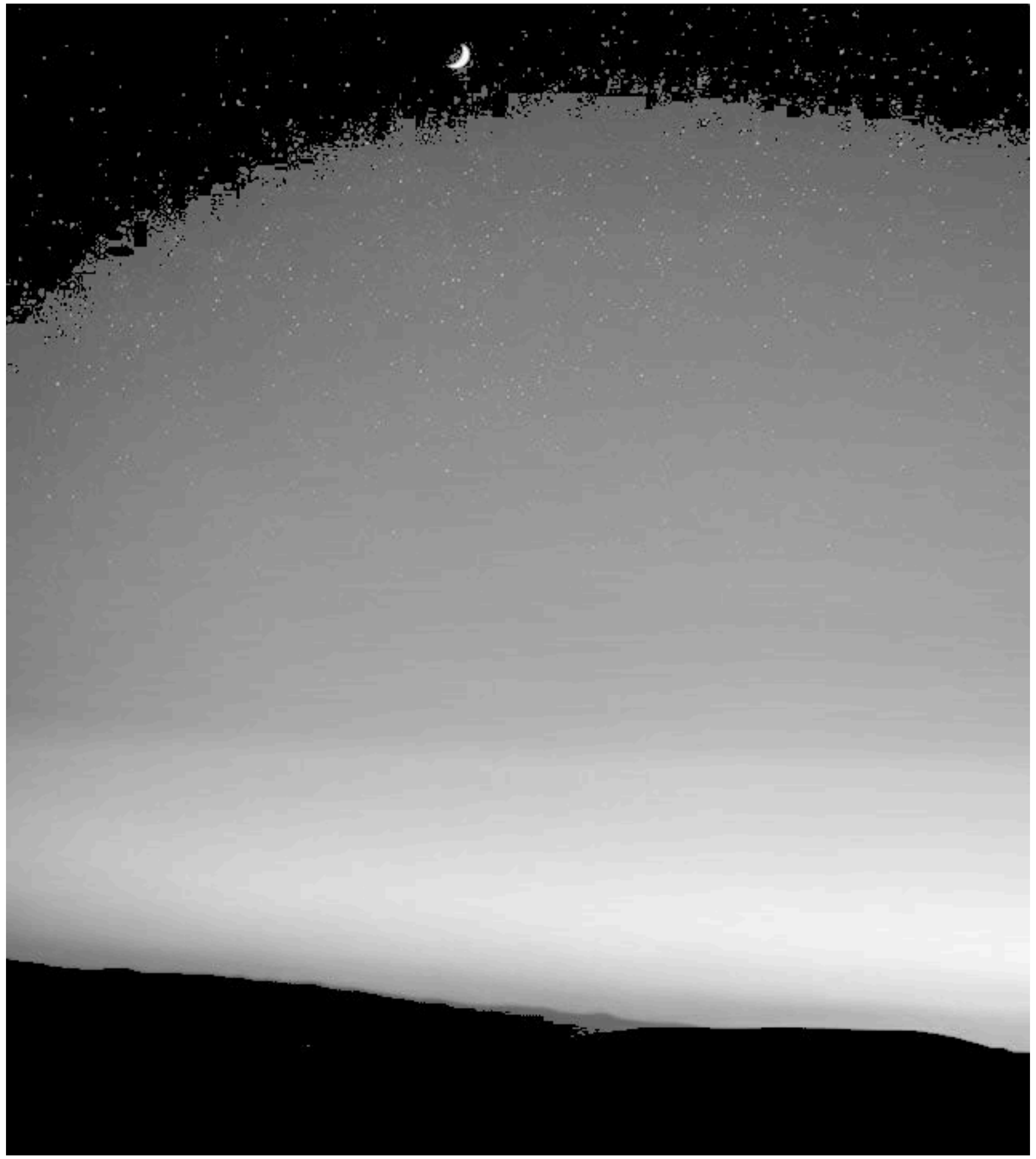
EXERCISE 1



In ComputerVision/ Exercise_01

Using loadPixels() mark the position of the moon with a red circle.

EXERCISE 1



Extract the position from the brightest pixel in the image using *brightness()*

EXERCISE 1

Finding the brightest value is quite difficult

- Webcams use auto exposure
- Various noise removal algorithms behind

EXERCISE 2

In ComputerVision/
Exercise_02

Find the average RGB value of video stream and represent it as a circle in the middle of the canvas.

Make the size of the circle bigger or smaller depending on the brightness.