Media: Video and Sound

p5.dom and p5.sound libraries

Capturing Video

Computer Vision

Playing Movies and Sounds

Chapter 13, Examples 13-1, 13-2, 13-3, 13-4

Today...

- We will see...
  - how to use libraries in p5.js
  - how to use video & sound
  - some examples of computer vision

Today...

- We will see...
  - **how to use libraries in p5.js**
  - how to use video & sound
  - some examples of computer vision

What is a library?

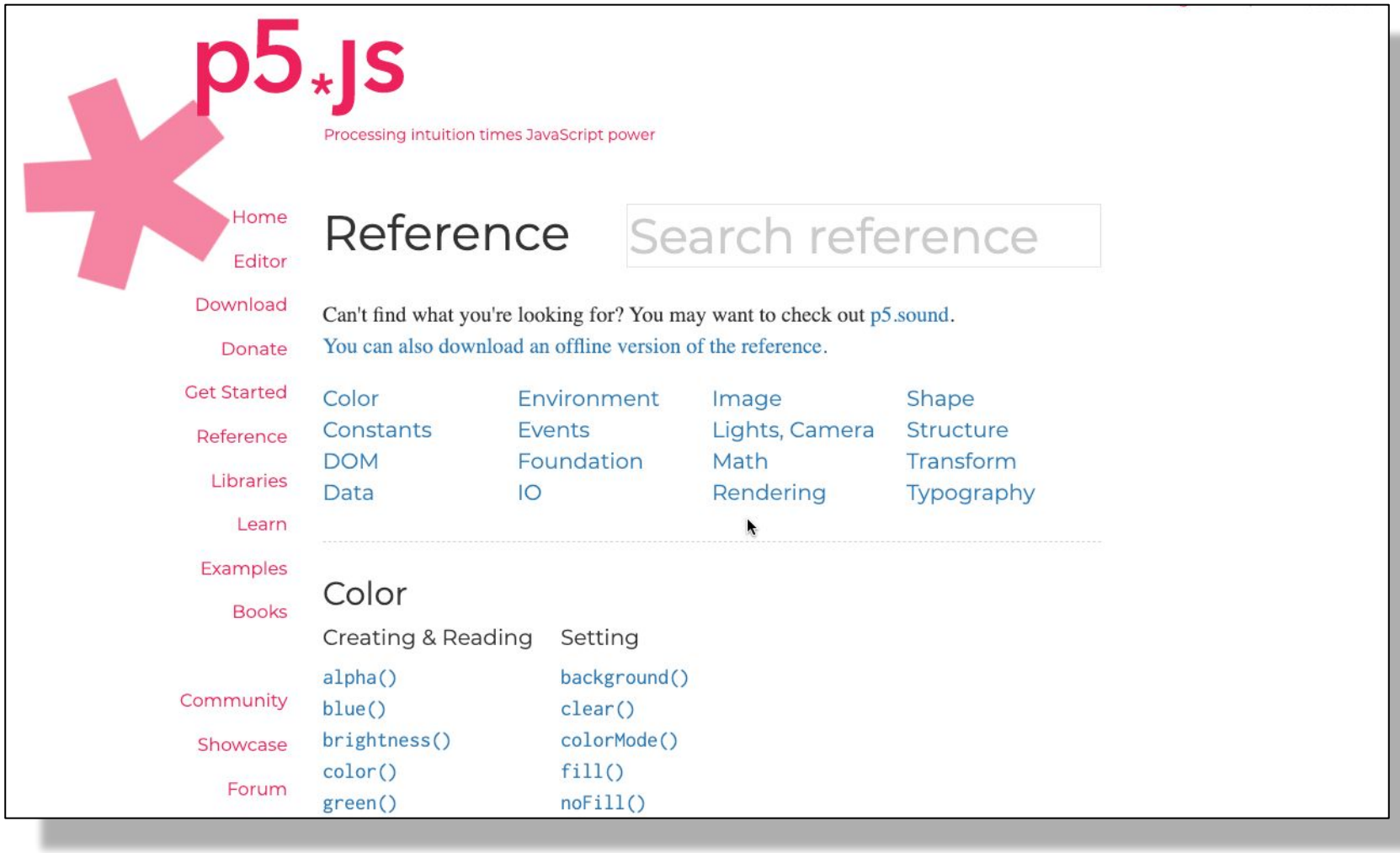
What is a library?

- A collection of pre-written code (e.g., functions, variables) that offers a particular functionality

What is a library?

https://p5js.org/

p5.js supports a set of drawing functionality



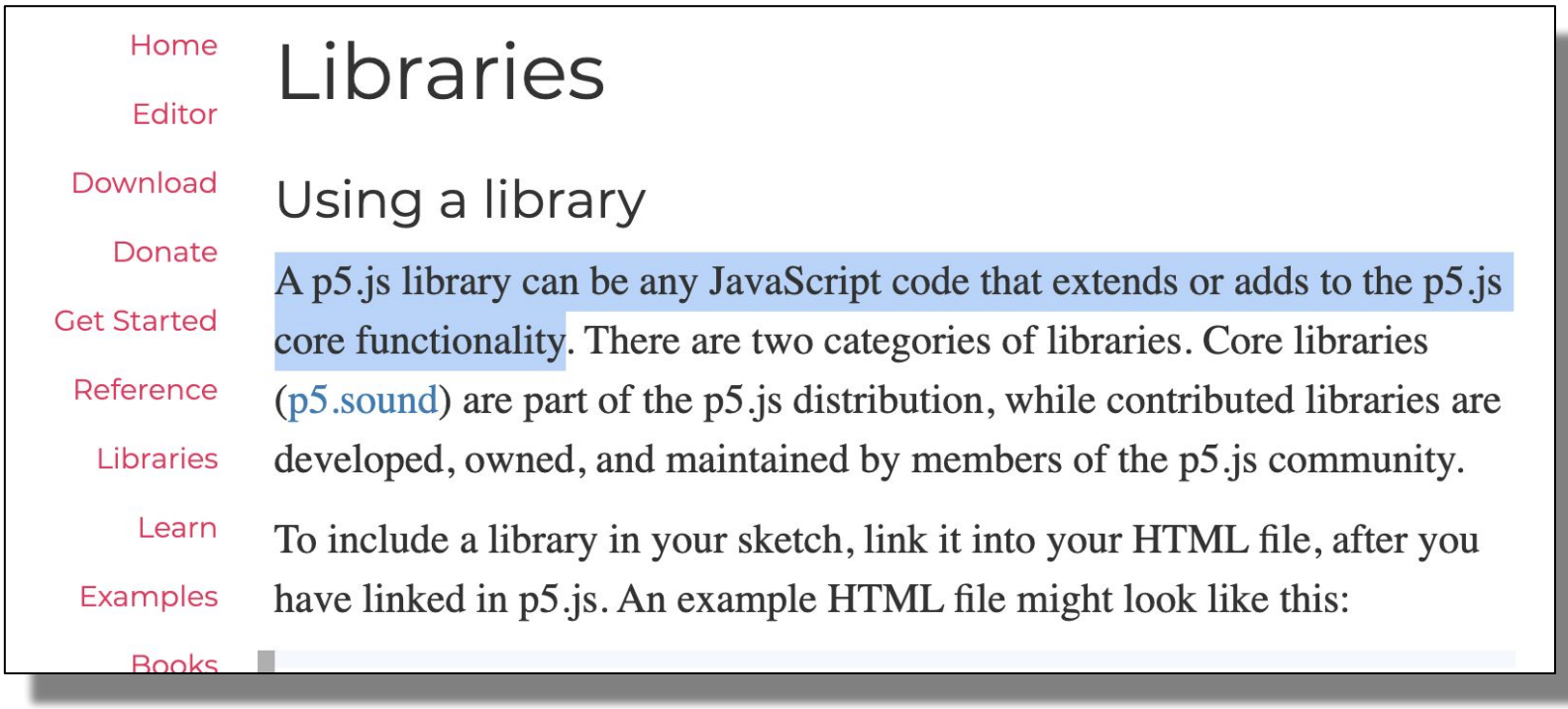
https://p5js.org/reference/

Why use libraries?

- To avoid rewriting the same code from scratch
  - same reason as writing functions

Why use libraries?

- To avoid rewriting the same code from scratch
  - same reason as writing functions
- To extend or add to the p5.js core functionality



Contributed p5 Libraries

- https://p5js.org/libraries/

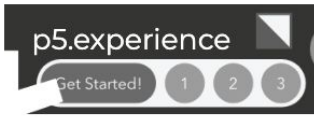
Core Libraries

p5.sound extends p5 with Web Audio functionality including audio input, playback, analysis and synthesis. Created by: Jason Sigal

p5.accessibility makes the p5 canvas more accessible to people who are blind and visually impaired.

Event driven, easy-to-use button library for p5.js. Created by: Martín del Río

CMYK ColorSpace Created by: JT Nimoy



Extensive library for p5.js that additional event-listening functionality for creating canv based web applications. Creat by: Felix Meichelböck

Community Libraries

p5.asciiart is a simple and easy to use image - to - ASCII art converter for p5js. Created by: Pawel Janicki

A Javascript library that enables communication between BLE devices and p5 sketches. Created by: Yining Shi, Jingwen Zhu, Tom Igoe

p5.collide2D provides tools for calculating collision detection for 2D geometry with p5.js. Created by: Ben Moren

Create animation loops with noise and GIF exports in one line of code. Created by: Peter Hayman

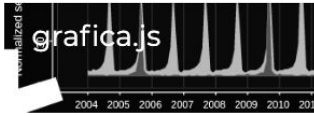
p5.geolocation provides techn for acquiring, watching, calculating, and geofencing us locations for p5.js. Created by Moren

p5.dimensions extends p5.js' vector functions to work in any number of dimensions. Created by: Smilebags, Max Segal

Simple 3D camera control with inertial pan, zoom, and rotate. Major contributions by Thomas Diewald. Created by: jWilliam Dunn

a library for making DOM manipulation simple Created by: Rohan Samra-O'Neill

With p5.bots you can interact with your Arduino (or other microprocessor) from within the browser. Use sensor data to drive a sketch; use a sketch to drive LEDs, motors, and more! Created by Created by: Sarah Groff-Palermo

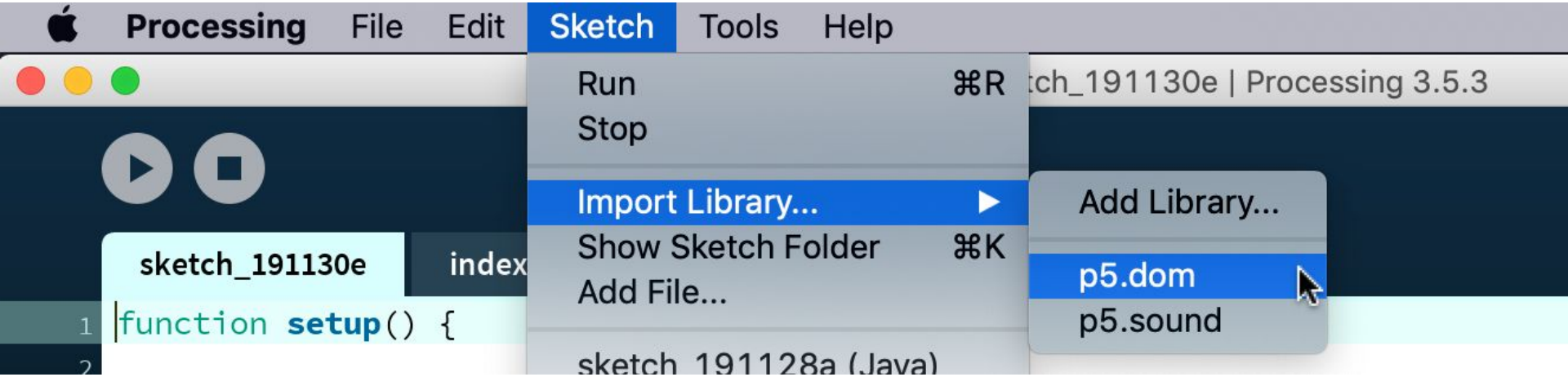


grafica.js lets you add simple highly configurable 2D plots t your p5.js sketches. Created b Javier Graciá Carpio

How to use libraries? (e.g., DOM library)

- libraries must be added ("imported") into your project

p5 web editor already includes "p5.dom" and "p5.sound" libraries, but you need to add them yourself in the Processing IDE



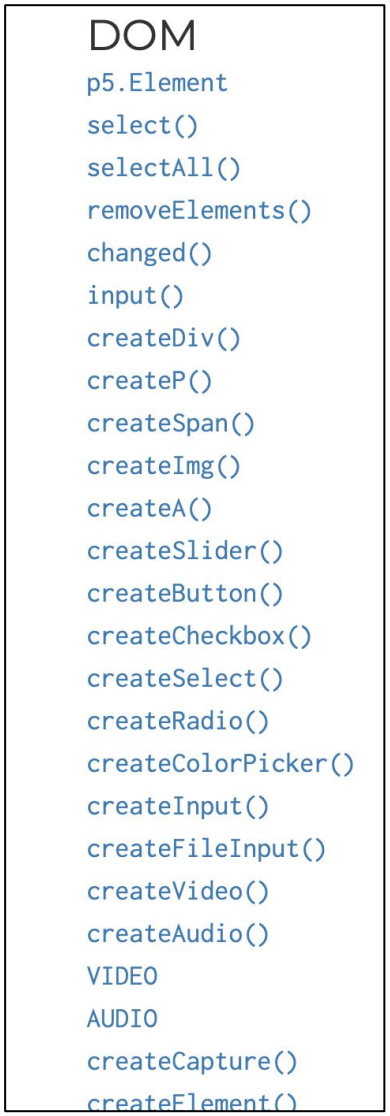
How to use libraries? (e.g., DOM library)

- make sure to add a link to it

```
<html>
<head>
  <script language="javascript" type="text/javascript"
src="libraries/p5.min.js"></script>
  <script language="javascript" type="text/javascript"
src="libraries/p5.dom.min.js"></script>
</head>
<body>
  <script src="sketch.js"></script>
</body>
```

How to use libraries? (e.g., DOM library)

- Every library has a reference explaining what it does & how to use it
- It offers HTML element generation functionality



https://p5js.org/reference/

Your Project is a Webpage

- Your p5.js script runs *inside* an HTML webpage
  - HTML code brings together your script, p5.js, and p5.js libraries

```
<html>
<head>
  <script language="javascript" type="text/javascript" src="libraries/p5.min.js"></script>
  <script language="javascript" type="text/javascript" src="libraries/p5.dom.min.js"></script>
</head>
<body>
  <script src="sketch.js"></script>
</body>
```

Today...

- We will see...
  - **how to use libraries in p5.js**
  - how to use video & sound
  - some examples of computer vision

Today...

- We will see...
  - how to use libraries in p5.js
  - **how to use video & sound**
  - some examples of computer vision

Using p5.dom Library to Capture Live Video

0) Make sure the p5.dom library is in your project

1) Declare a global variable for a *video Element object*

```
let camera;
```

2) Create the *video Element object* in setup()

```
camera = createCapture(VIDEO);
camera.size(width, height);
```

3) Use the video Element object like a p5 Image

- Draw the current video frame on the canvas:

```
image(camera, 0, 0);
```

- Get the colour at a pixel location:

```
camera.get(x, y);
```

- Process the whole pixel array:

```
camera.pixels[i];
```

video-capture

```
let camera;

void setup() {
  createCanvas(320, 240);
  camera = createCapture(VIDEO);
  camera.size(width, height);
  camera.hide();
}

function draw() {
  image(camera, 0, 0);
}
```

this hides the associated HTML video element

https://editor.p5js.org/sanghosuh/sketches/YqyJacyqm

video-painterly

using .get(x, y) with video frame

```
for (let i = 0; i < 100; i++) {
  let x = random(width);
  let y = random(height);
  stroke(camera.get(x, y));
  let s = random(1, 5);
  strokeWeight(s);
  point(x, y);
}
```

gets the colour of video pixel

https://editor.p5js.org/sanghosuh/sketches/FZW9WdKdY

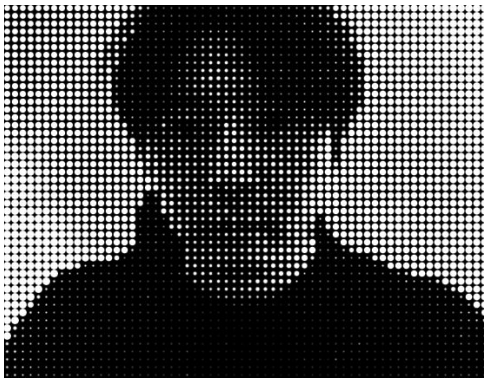
video-grid

```
// loop through all grid positions
for (let x = maxSize / 2; x < width; x += maxSize) {
  for (let y = maxSize / 2; y < height; y += maxSize) {

    // get the colour at the corresponding pixel
    let vx = floor(map(x, 0, width - 1, 0, camera.width - 1));
    let vy = floor(map(y, 0, height - 1, 0, camera.height - 1));
    let pixelColour = camera.get(vx, vy);

    // get brightness and convert it to a size
    let b = brightness(pixelColour);
    let s = map(b, 0, 100, 0, maxSize * 1);

    fill(255);
    ellipse(x, y, s, s);
  }
}
```



https://editor.p5js.org/sanghosuh/sketches/_3lAXYUCI

video-processing

```
image(camera, 0, 0);

loadPixels();
for (let x = 0; x < width; x++) {
  for (let y = 0; y < height; y++) {

    // index into pixels array
    let i = (x + y * width) * 4;

    // extract red, green, blue, alpha
    let r = pixels[i];
    let g = pixels[i + 1];
    let b = pixels[i + 2];
    let a = pixels[i + 3];

    // process pixel here ...
```

https://editor.p5js.org/sanghosuh/sketches/BdS_uw6kr

Using p5.dom Library to Play Video

0) Make sure the “p5.dom” library is in your project

1) Add a video file to your sketch’s data folder.

2) Declare a global variable for a *video Element object*

```
let movie;
```

3) Create the *video Element object* in `preload()`

```
movie = createVideo("data/flyboard.mp4");
```

4) Start playback in `setup()`

```
movie.play();
```

5) Use the *video Element object* like a p5 Image

```
image(movie, 0, 0);
...
```

movie-play

```
let movie;

function preload() {
  movie = createVideo("data/flyboard.mp4");
}

function setup() {
  createCanvas(640, 360);
  movie.size(width, height);
  movie.hide();
  movie.play();
}

function draw() {
  image(movie, 0, 0);
}
```

https://editor.p5js.org/cs105/sketches/7BZWvfcfs

(detect if movie is done playing)

```
// check if video is over
if (movie.time() >= movie.duration()) {

   ...

}
```

returns the time of the current movie frame in seconds

returns the length of the whole movie in seconds

https://www.w3schools.com/tags/ref_av_dom.asp

movie-scrub

```
if (movie.duration() > 0) {
  let t = map(mouseX, 0, width, 0, movie.duration());
  movie.time(t);
}
```

https://editor.p5js.org/cs105/sketches/ZbuoreaRW

Using the p5.sound Library to Play Sounds

0) Add a sound file to your sketch's data folder.

1) Declare a global variable for the *SoundFile* object

```
let honk;
```

2) Load the sound file into the variable in preload()

```
honk = loadSound("data/honk.wav");
```

3) Play sound when you want

```
honk.play();
```

Starter: https://editor.p5js.org/cs105/sketches/NyM_n_1QE

sound

```
let honk;
let horn;

function preload() {
  // load sound files from data directory
  honk = loadSound("data/honk.wav");
  horn = loadSound("data/horn.wav");
}

function mousePressed() {
  if (mouseX < 50) {
    honk.play();
  } else {
    horn.play();
  }
}
```

https://editor.p5js.org/cs105/sketches/WSsEP2wUB

Today...

- We will see...
  - how to use libraries in p5.js
  - **how to use video & sound**
  - some examples of computer vision

Today...

- We will see...
  - how to use libraries in p5.js
  - how to use video & sound
  - **some examples of computer vision**

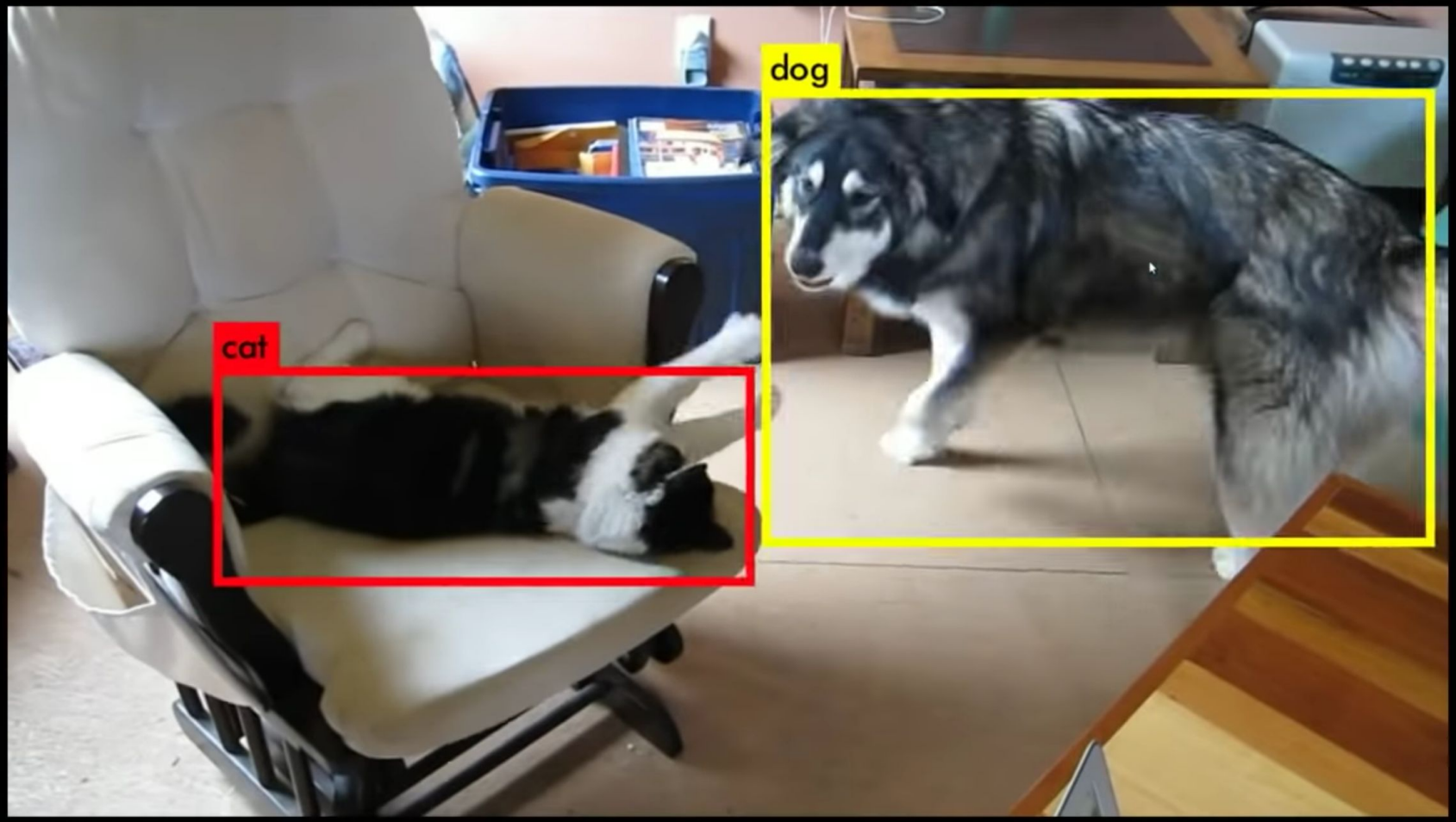
Computer Vision

- "Computer vision" refers to a broad class of algorithms that allow computers to make intelligent assertions about digital images and video (Levin, 2006)

- Computers that can "see"
- Using the camera as a "sensor"

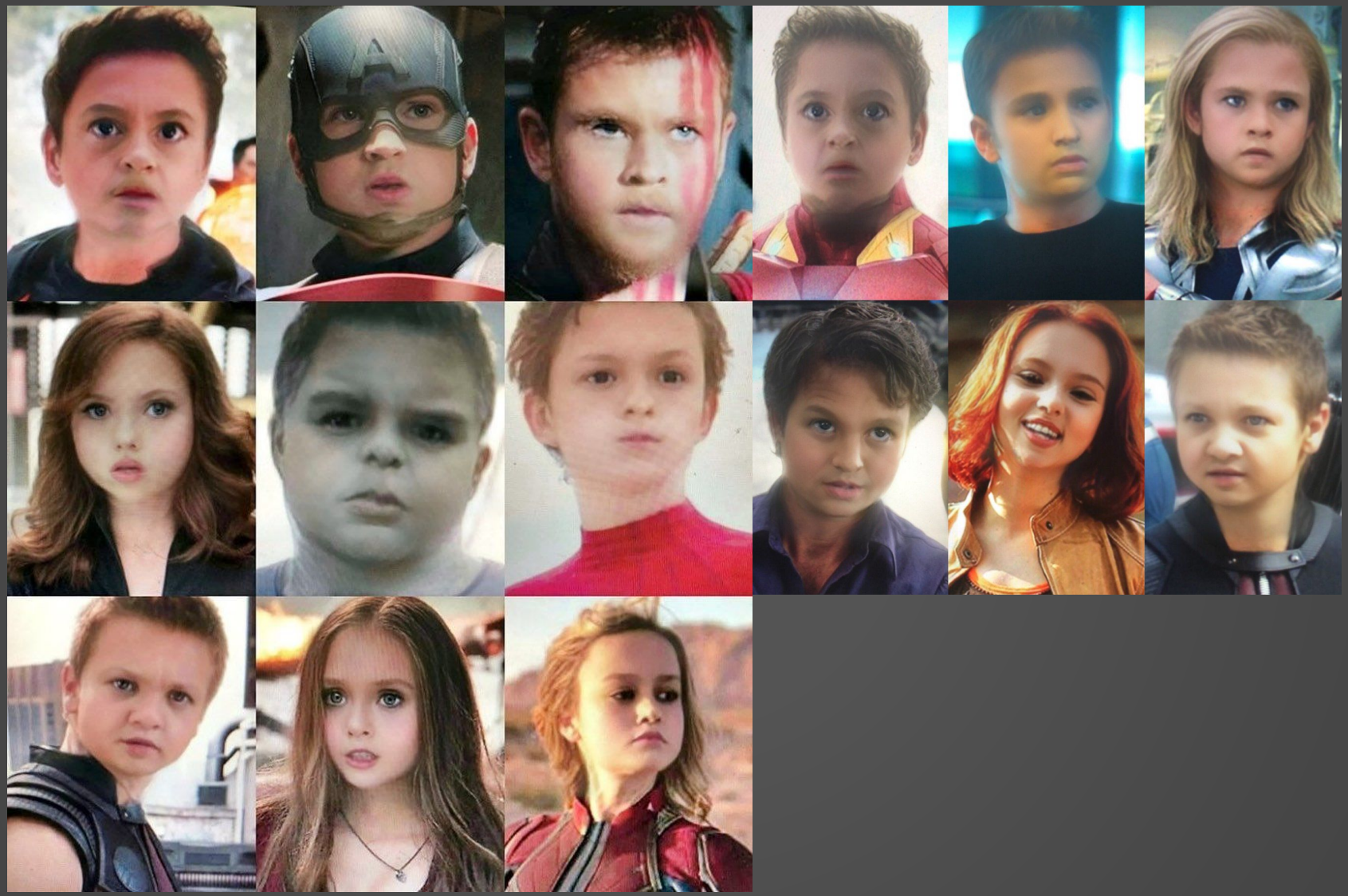
VIDEOPLACE Mini-documentary (1988)

- https://youtu.be/dmmxVA5xhuo?t=133



https://youtu.be/Cgxsv1rjJhI?t=271

https://youtu.be/BKlOqbx3sbU?t=332



CREDIT: Know Your Meme

https://www.youtube.com/watch?v=bPhUhypV27w

A Simple Computer Vision Algorithm

```
for each video frame:
    identify a "special" pixel
    based on some criteria
    use that pixel's location
    to control the computer
```

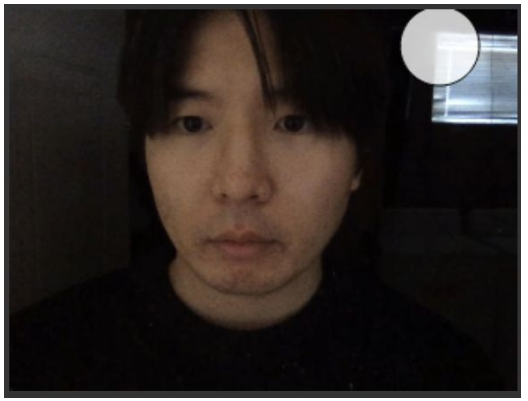
This is just a special kind of array operation, ...
very similar to finding the largest element in an array.

Array Operation: Find Largest Element Value

- Search the array to find the largest element value
  - Start by guessing that the largest value ss the first element
  - check all other elements one-by-one to see if any are larger
  - if larger value is found, it becomes new largest value found so far

```
// find the largest element value
// (assuming arr has at least length 1)
let largest = arr[0];
for (let i = 1; i < arr.length; i++) {
  if (arr[i] > largest) {
    largest = arr[i];
  }
}
print("largest:", largest);
```

bright-track



```
let brightest = 0;
let brightestLoc = [0, 0];

for (let x = 0; x < width; x++) {
  for (let y = 0; y < height; y++) {

    let i = (x + y * width) * 4;
    let r = pixels[i];
    let g = pixels[i + 1];
    let b = pixels[i + 2];
    let bright = brightness(color(r, g, b));

    if (bright > brightest) {
      brightest = bright;
      brightestLoc = [x, y];
    }
  }
}
```

https://editor.p5js.org/cs105/sketches/RJQqBip9o

colour-track

```
let best = 255 * 255 * 255;
let bestLoc = [0, 0];

for (let x = 0; x < width; x++) {
  for (let y = 0; y < height; y++) {
    let i = (x + y * width) * 4;
    let r = pixels[i];
    let g = pixels[i + 1];
    let b = pixels[i + 2];

    // t is the target colour to find best match
    let d = dist(red(t), green(t), blue(t), r, g, b);

    if (d < best) {
      best = d;
      bestLoc = [x, y];
    }
  }
}
```

https://editor.p5js.org/sanghosuh/sketches/mK3oxQuyB

chroma-key


```
let target;
let sensitivity = 30;

for (let x = 0; x < width; x++) {
  for (let y = 0; y < height; y++) {

    ...

    // t is the target colour to find best match
    let d = dist(red(t), green(t), blue(t), r, g, b);

   if (d < sensitivity) {
          pixels[i] = 255;
          pixels[i + 1] = 0;
          pixels[i + 2] = 0;
       }
    }
  }
}
```

https://editor.p5js.org/sanghosuh/sketches/aWIQdUhKf

https://evaluate.uwaterloo.ca/