

Algorithmic
Intelligence

Module B

Unit #10

CoCo SSD

Bounding Box

Video

Classification





Module B Unit #10 CoCoSSD bounding box video classification

Sketch B10.1	index.html
Sketch B10.2	getting the video stream
Sketch B10.3	now to detect
Sketch B10.4	drawing the bounding box
Sketch B10.5	labelling



Introduction to coco ssd bounding box video classification

In the last unit, we looked at a single image. Now we are going to see how well it works with a live stream video from your webcam or other device.



Sketch B10.1 index.html

Make sure you have the `ml5.js` line of code in your `index.html` file.

```
<!DOCTYPE html>
<html lang="en"><head>
  <script src="https://cdn.jsdelivr.net/npm/p5@2.2.2/lib/p5.js"></script>
  <script src="https://unpkg.com/ml5@1/dist/ml5.min.js"></script>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="utf-8">
</head>
<body>
  <main>
  </main>
  <script src="sketch.js"></script>
</body></html>
```



Sketch B10.2 getting the video stream

First, set up a video stream from the webcam.

```
let video
let detector

async function setup()
{
  createCanvas(640, 480)
  video = await createCapture(VIDEO)
  video.size(width, height)
  video.hide()
  detector = await ml5.objectDetection("cocossd")
}

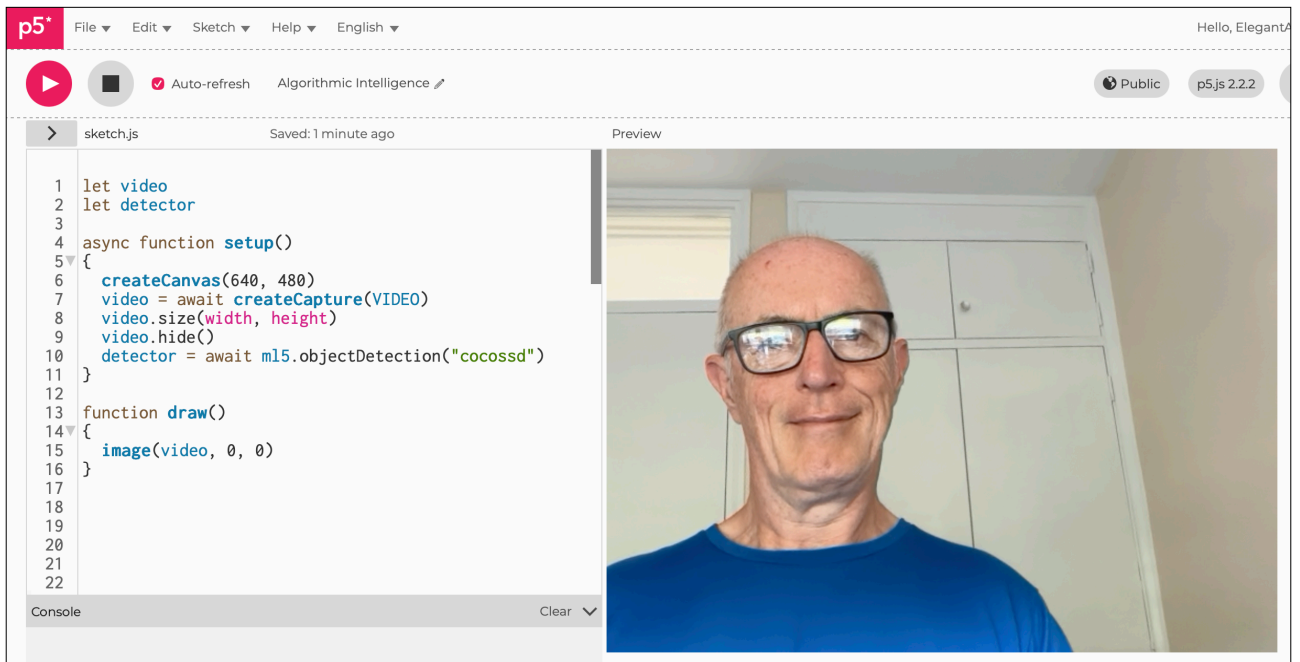
function draw()
{
  image(video, 0, 0)
}
```



Notes

You should have a video image; you may have to allow access.

Figure B10.2





Sketch B10.3 now to detect

Let's start detecting something or somebody.

```
let video
let detector

async function setup()
{
  createCanvas(640, 480)
  video = await createCapture(VIDEO)
  video.size(width, height)
  video.hide()
  detector = await ml5.objectDetection("cocossd")
  detector.detectStart(video, gotDetections)
}

function draw()
{
  image(video, 0, 0)
}

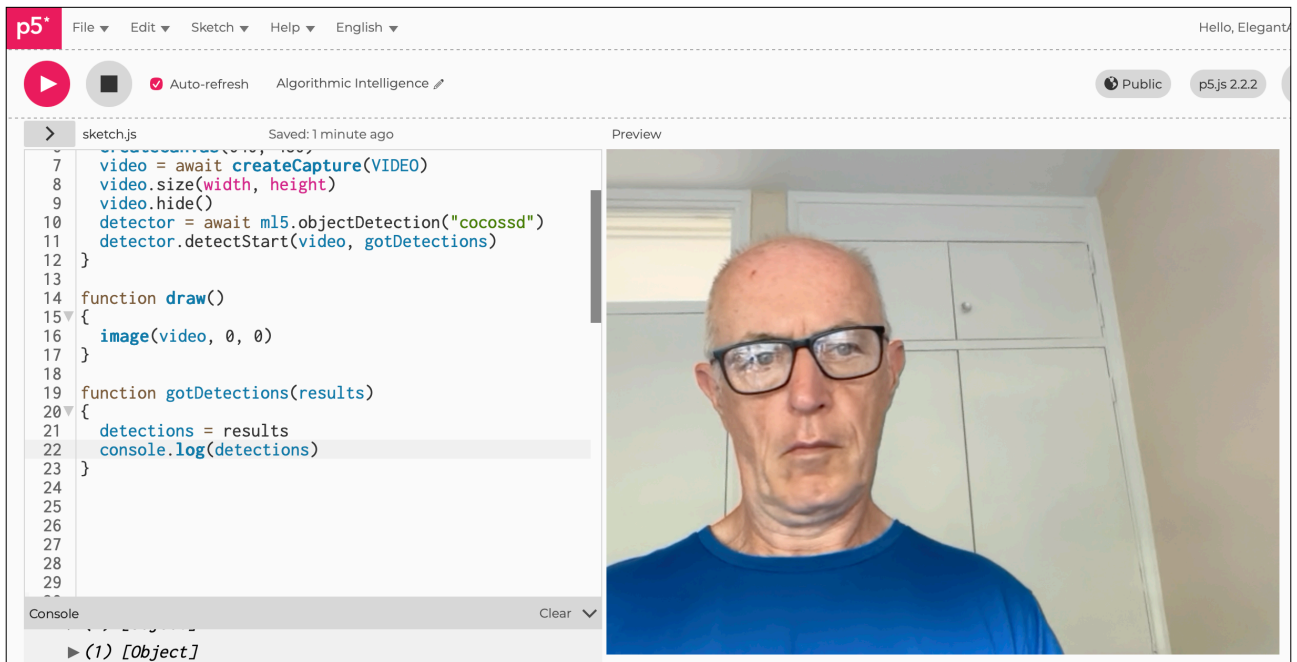
function gotDetections(results)
{
  detections = results
  console.log(detections)
}
```



Notes

You may get more than one object detected depending on what is in the background. You can pause the sketch and take a look in the console.

Figure B10.3





Sketch B10.4 drawing the bounding box

The bounding box is just as before, although this time it moves with the object or comes and goes.

```
let video
let detector
let det

async function setup()
{
  createCanvas(640, 480)
  video = await createCapture(VIDEO)
  video.size(width, height)
  video.hide()
  detector = await ml5.objectDetection("cocossd")
  detector.detectStart(video, gotDetections)
}

function draw()
{
  image(video, 0, 0)
  for (let i = 0; i < detections.length; i++)
  {
    det = detections[i]
    stroke('green')
    strokeWeight(2)
    noFill()
    rect(det.x, det.y, det.width, det.height)
  }
}

function gotDetections(results)
{
  detections = results
  console.log(detections)
}
```



Notes

You should have a bounding box, and if you hold something up (like a mobile phone), it should clock that as well.

Figure B10.4

The screenshot shows the p5.js IDE interface. The code editor on the left contains the following JavaScript code:

```
1 let video
2 let detector
3 let det
4
5 async function setup()
6 {
7   createCanvas(640, 480)
8   video = await createCapture(VIDEO)
9   video.size(width, height)
10  video.hide()
11  detector = await ml5.objectDetection("cocossd")
12  detector.detectStart(video, gotDetections)
13 }
14
15 function draw()
16 {
17   image(video, 0, 0)
18   for (let i = 0; i < detections.length; i++)
19   {
20     det = detections[i]
21     stroke('green')
22     strokeWeight(2)
23     noFill()
24     rect(det.x, det.y, det.width, det.height)
25   }
26 }
```

The preview window on the right shows a video feed of a man in a blue shirt and glasses holding a black smartphone. A green bounding box is drawn around the smartphone, indicating object detection. The IDE interface includes a menu bar (File, Edit, Sketch, Help, English), a toolbar with a play button, a stop button, and a checkmark for 'Auto-refresh', and a console at the bottom showing the output: `▶ (2) [Object, Object]`.



Sketch B10.5 labelling

We can put the labels in, even though you could stop/pause the programme and look inside the array.

```
let video
let detector
let det

async function setup()
{
  createCanvas(640, 480)
  video = await createCapture(VIDEO)
  video.size(width, height)
  video.hide()
  detector = await ml5.objectDetection("cocossd")
  detector.detectStart(video, gotDetections)
}

function draw()
{
  image(video, 0, 0)
  for (let i = 0; i < detections.length; i++)
  {
    det = detections[i]
    stroke('green')
    strokeWeight(2)
    noFill()
    rect(det.x, det.y, det.width, det.height)
    noStroke()
    fill('white')
    textSize(24)
    text(det.label, det.x + 10, det.y + 24)
  }
}

function gotDetections(results)
{
```

```
detections = results
console.log(detections)
}
```

Notes

There is a limit to the number of objects it has been trained on, so you may well get some interesting labelling.

Challenge

Add the confidence score after label.

Figure B10.5

