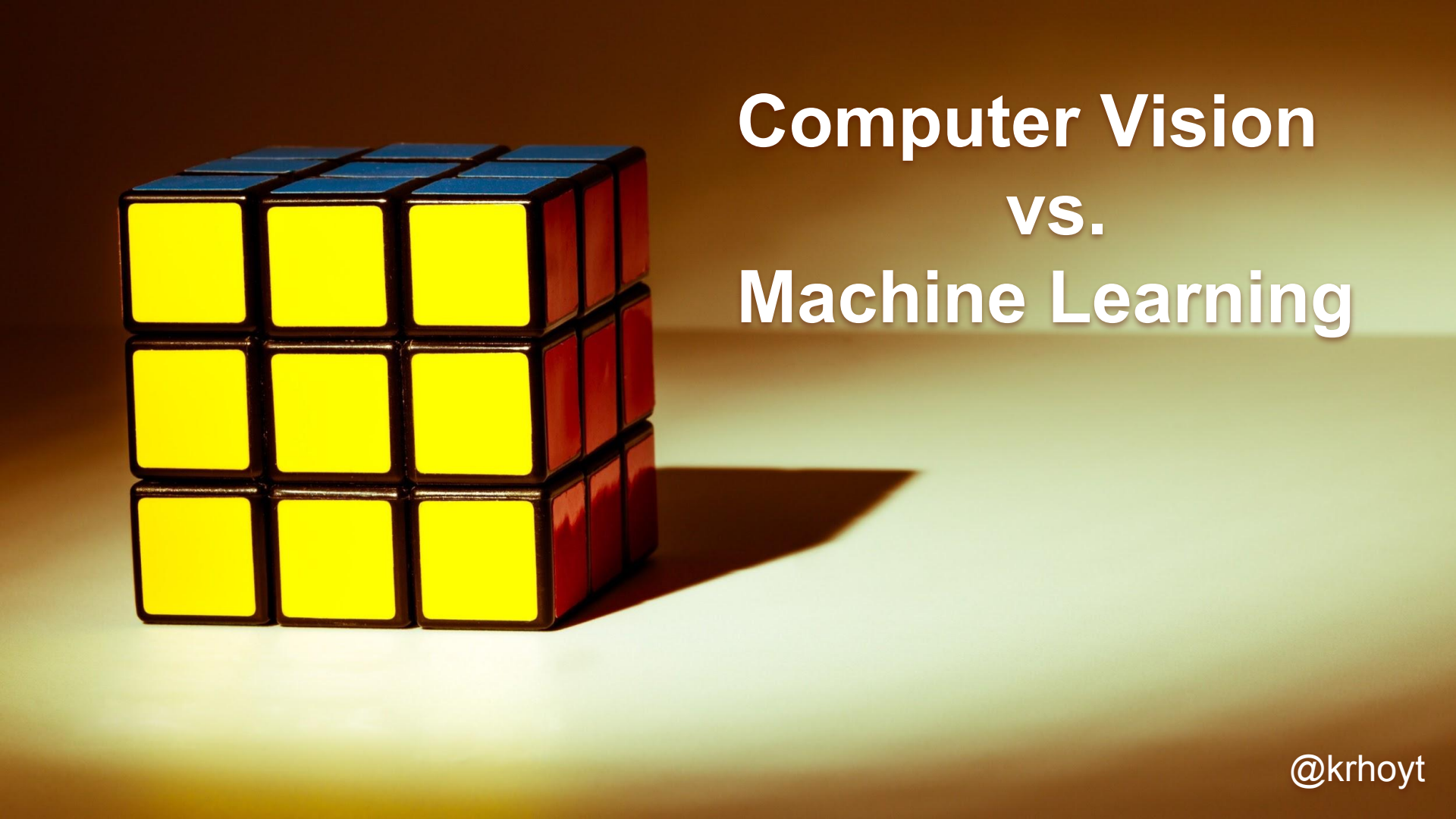

Computer Vision
vs.
Machine Learning
@krhoyt

WebRTC lands in Safari iOS as of 11.2!

```
<video playsinline></video>
<canvas width="640" height="480"></canvas>
```

```
this.video = document.querySelector( 'video' );

navigator.mediaDevices.getUserMedia( {
  audio: false,
  video: true
} ).then( ( stream ) => {
  this.video.srcObject = stream;
  this.video.play();
} ).catch( ( error ) => {
  console.error( error );
} );
```

```
this.canvas = document.querySelector( 'canvas' );
this.context = this.canvas.getContext( '2d' );

// ...

this.context.drawImage( this.video, 0, 0, 640, 480 );

// [r, g, b, a, ...]
let pixels = this.context.getImageData( 0, 0, 640, 480 );
```

```
for( let p = 0; p < pixels.data.length; p += 4 ) {
  // let average = ( r + g + b ) / 3;
  let brightness = (
    ( 0.299 * pixels.data[p] ) +
    ( 0.587 * pixels.data[p + 1] ) +
    ( 0.114 * pixels.data[p + 2] ) );

  pixels.data[p] = brightness;
  pixels.data[p + 1] = brightness;
  pixels.data[p + 2] = brightness;
}

this.context.putImageData( pixels, 0, 0 );
```

```
// Grayscale
jsfeat.imgproc.grayscale(
  this.pixels.data, this.canvas.clientWidth,
  this.canvas.clientHeight, this.image
);

// Gaussian blur
let kernel = ( 3 + 1 ) << 1;
jsfeat.imgproc.gaussian_blur(
  this.image, this.image,
  kernel, 0
);
```

```
.filtered {
  filter: grayscale( 1 ) blur( 3px );
}
```

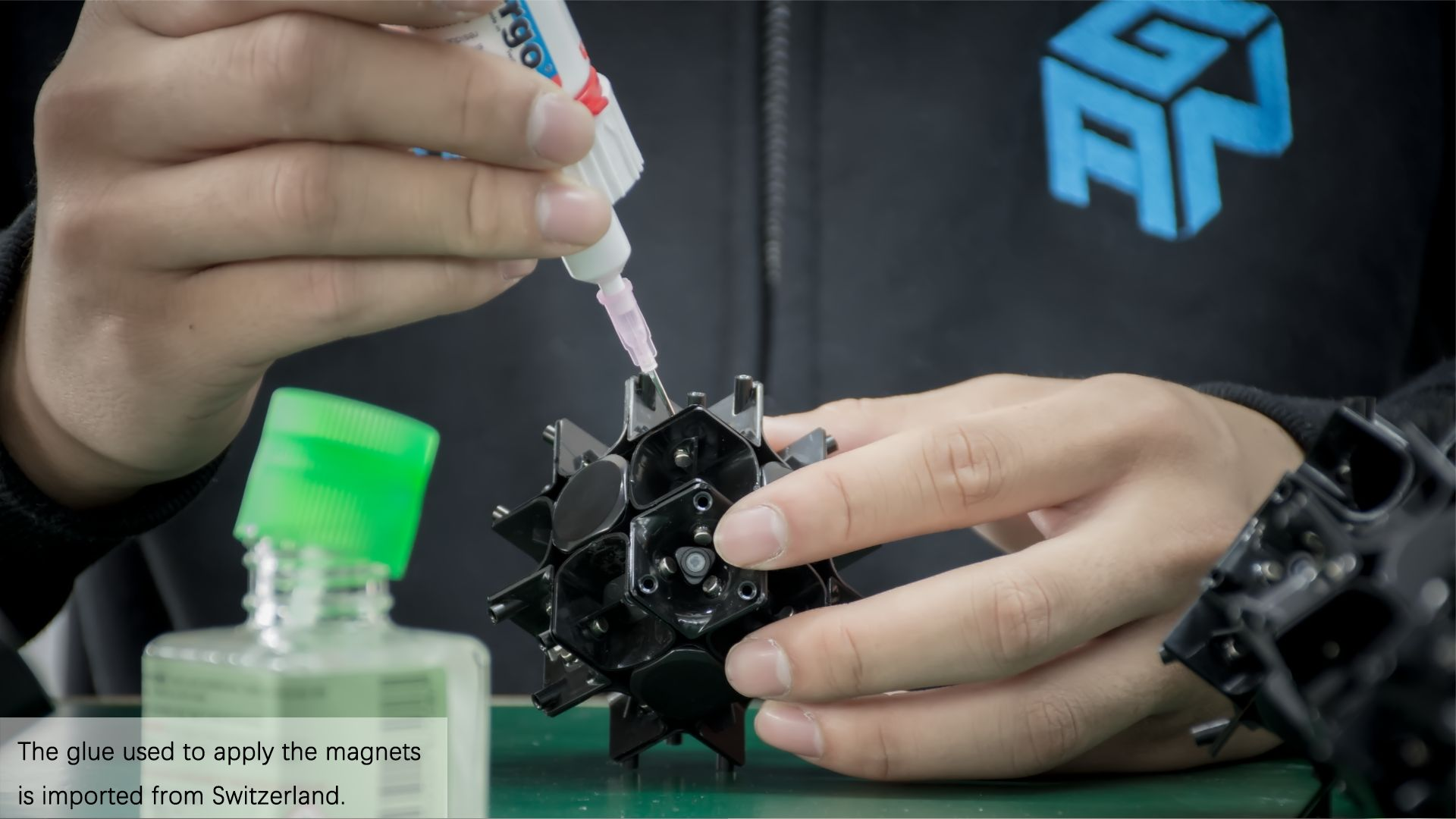
```
// Canny edge detection
jsfeat.imgproc.canny( this.image, this.image, 20, 40 );

// Emphasize edges
jsfeat.imgproc.dilate( this.image, this.image );
```

```
// Contours
this.contours = CV.findContours( this.image, [] );
```

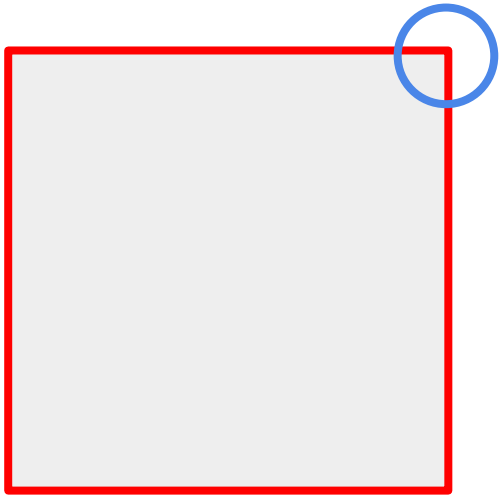
```
for( let c = 0; c < this.contours.length; c++ ) {
  // Epsilon (variation) based on length of contour array
  this.contours[c] = CV.approxPolyDP(
    this.contours[c], this.contours[c].length * 0.03
  );
}
```

43,252,003,274,489,856,000
(43 quintillion)


4.737
SPEED STACKS

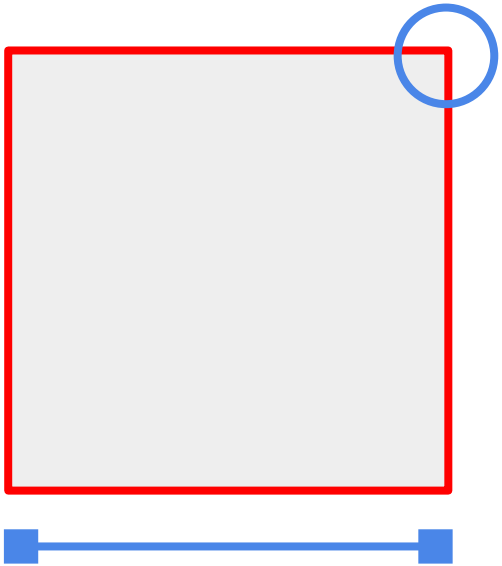


The glue used to apply the magnets is imported from Switzerland.



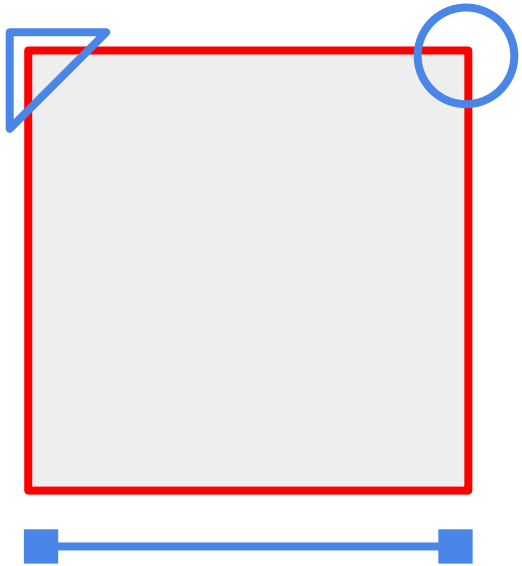
- Four corners

```
polygon.length === 4
```



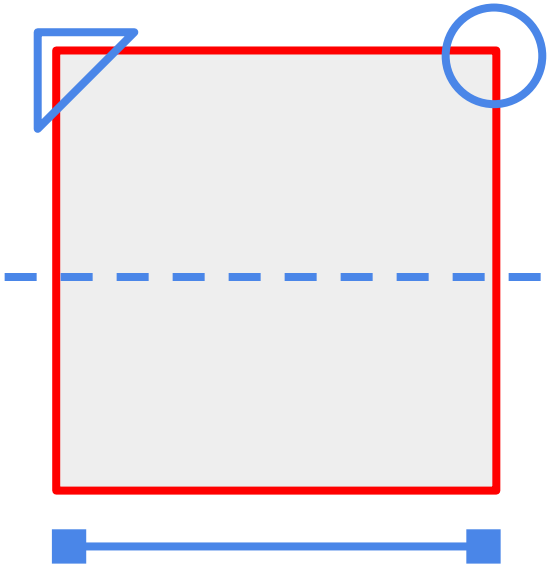
- Four corners
- Equal length sides

```
let top =
  polygon[1].x - polygon[0].x;
let right =
  polygon[2].y - polygon[1].y;

if( top === right ) { // ... }
```



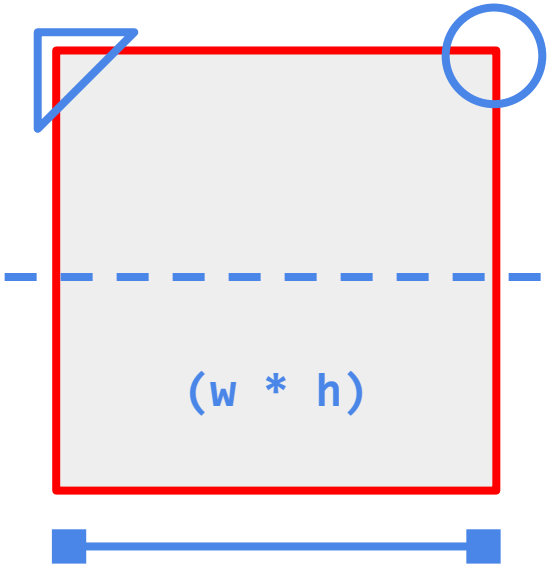
- Four corners
- Equal length sides
- Ninety degree angles

```
let angle = Math.atan2(
  polygon[1].y - polygon[0].y,
  polygon[1].x - polygon[0].x
) * ( 180 / Math.PI );
```



- Four corners
- Equal length sides
- Ninety degree angles
- Minimal rotation

```
let rotation = Math.atan2(
  polygon[1].y - polygon[0].y,
  polygon[1].x - polygon[0].x
) * ( 180 / Math.PI );
```



- Four corners
- Equal length sides
- Ninety degree angles
- Minimal rotation
- Similar surface areas

```
let width =
  polygon[1].x - polygon[0].x;
let height =
  polygon[2].y - polygon[1].y;

let area = width * height;
```

{r: 186, g: 12, b: 47}

{r: 0, g: 154, b: 68}

{r: 0, g: 61, b: 165}

{r: 254, g: 80, b: 0}

{r: 255, g: 215, b: 0}

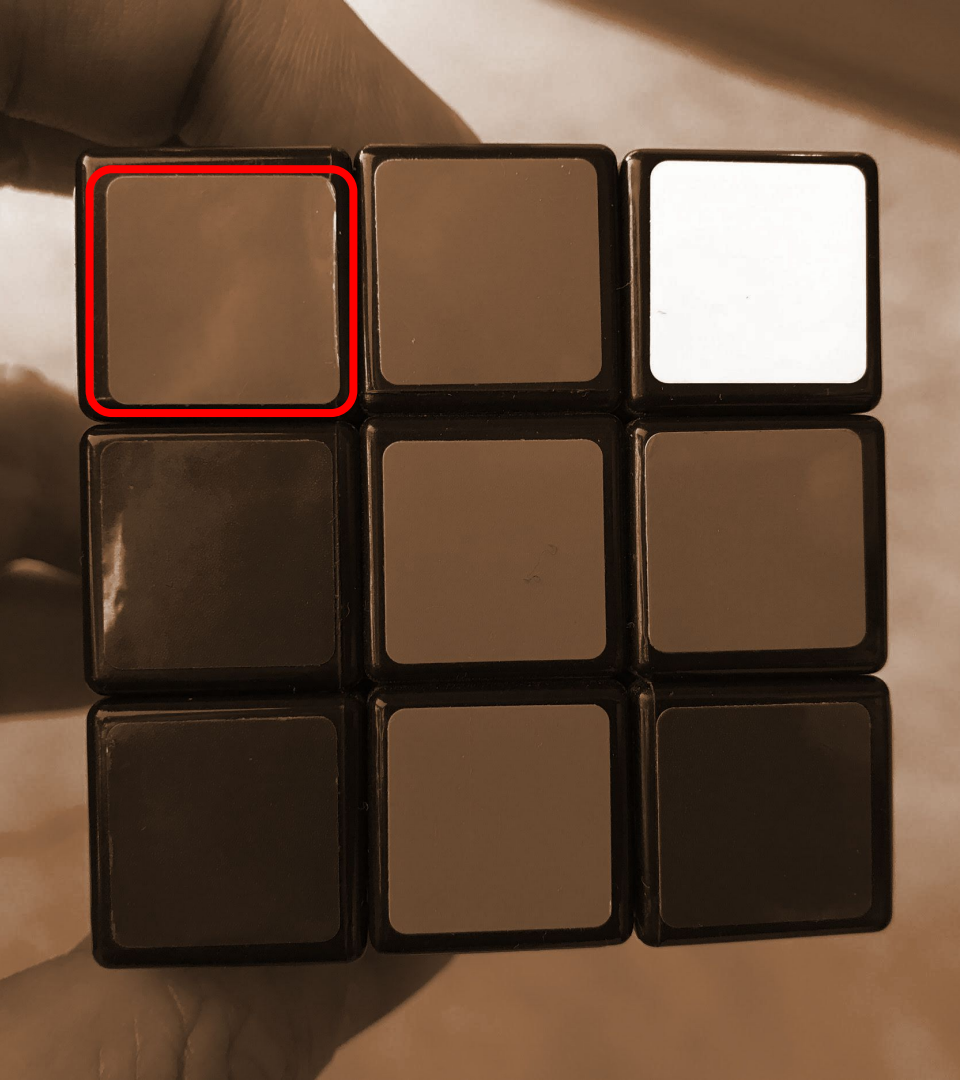
{r: 255, g: 255, b: 255}

{r: 68, g: 182, b: 112}

{r: 0, g: 154, b: 68}

$$\sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$$

```
Math.sqrt(
  Math.pow( ( 68 - 0   ), 2 ) +
  Math.pow( (182 - 154), 2 ) +
  Math.pow( (112 - 68 ), 2 )
) === 85.697
```



{r: 127, g: 90, b: 61}

{r: 0, g: 154, b: 68}

$$\sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$$

```
Math.sqrt(
  Math.pow( (127 - 0  ), 2 ) +
  Math.pow( ( 90 - 154), 2 ) +
  Math.pow( ( 61 - 68 ), 2 )
) === 142.387
```


POWER7
POWER7
POWER7
WATSON
POWER7
POWER7
POWER7


Department of
Information Technology,
Government of Goa

```
curl -X POST WATSON_URL
     -u apikey:WATSON_KEY
     -F images_file=@horea.porutiu.jpg
```

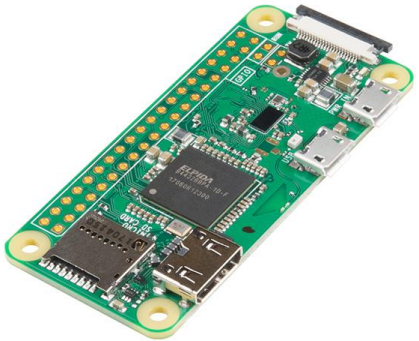
```
{
  "images": [{
    "classifiers": [{
      "classifier_id": "default",
      "name": "default",
      "classes": [{
        "class": "orator",
        "score": 0.772,
        "type_hierarchy": "/person/orator"
      }, ...]
    }],
    "image": "horea.porutiu.jpg"
  }],
  "images_processed": 1,
  "custom_classes": 0
}
```

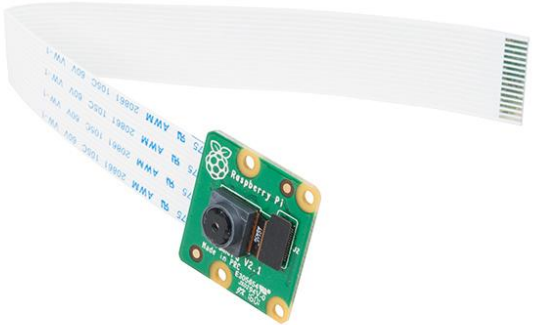
```
this.faces = [];

// Setup tracking
this.tracker = new tracking.ObjectTracker( 'face' );
this.tracker.setInitialScale( 4 );
this.tracker.setStepSize( 2 );
this.tracker.setEdgesDensity( 0.1 );

// Start tracking
tracking.track( this.video, this.tracker, {camera: true} );

// Capture detected faces
this.tracker.on( 'track', ( evt ) => {
  this.faces = evt.data.slice( 0 );
} );
```

```
curl -X POST WATSON_URL
     -u apikey:WATSON_KEY
     -F name=faces
     -F Kevin_positive_examples=@Kevin_positive_examples.zip
     -F Paige_positive_examples=@Paige_positive_examples.zip
```

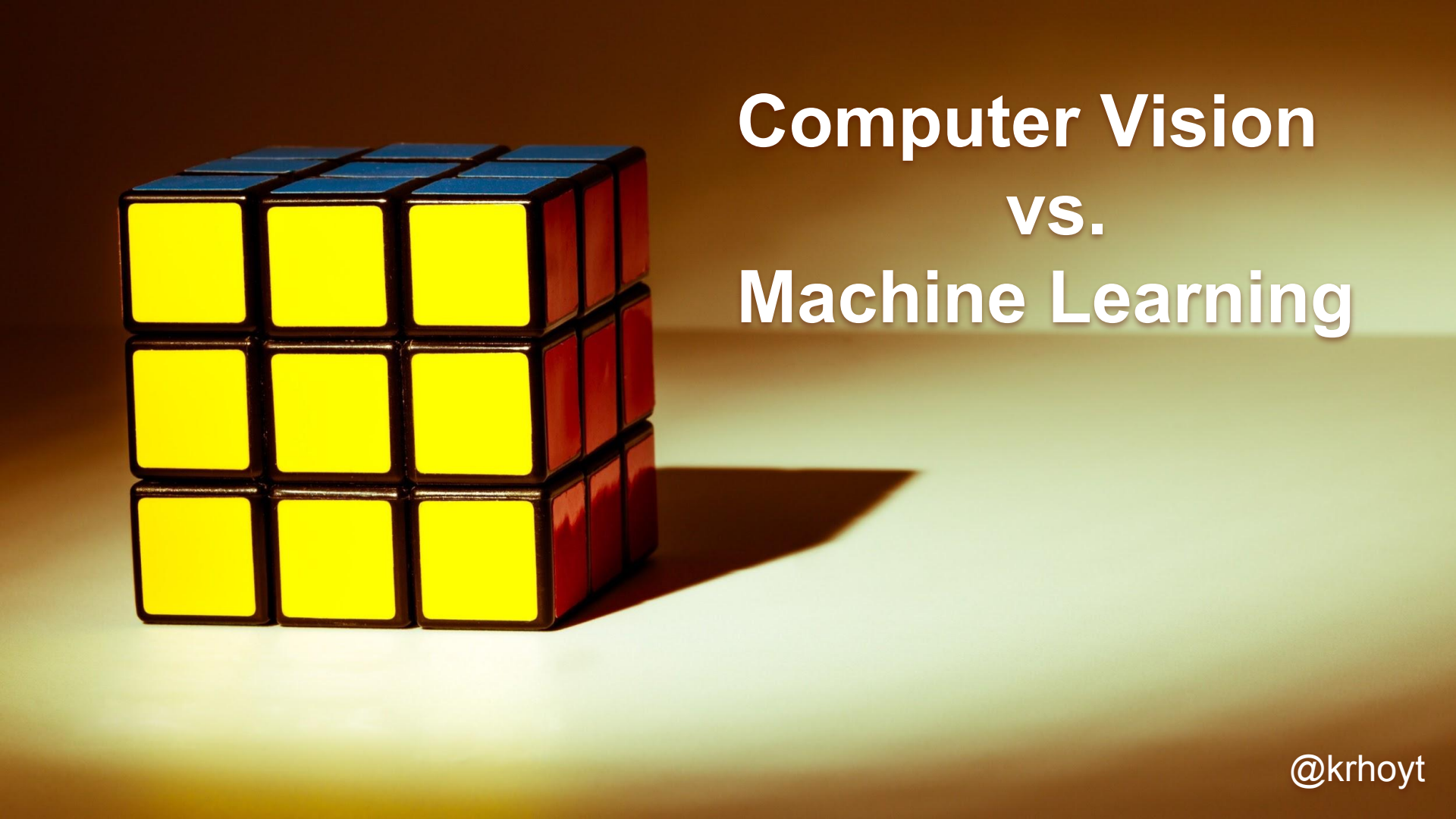
```
{
  "classifier_id": "faces_1568032002",
  "name": "faces",
  "status": "ready",
  "owner": "eb22ec50-1f7b-406e-86b9-ab5bdaafa82b",
  "created": "2018-09-05T22:25:04.666Z",
  "updated": "2018-09-05T22:25:04.666Z",
  "classes": [
    {"class": "Paige"},
    {"class": "Kevin"}
  ],
  "core_ml_enabled": true
}
```

```
curl -X POST WATSON_URL
     -u apikey:WATSON_KEY
     -F classifier_ids=faces_1568032002
     -F images_file=@horea.porutiu.jpg
```




Raspberry Pi

@krhoyt


Computer Vision
vs.
Machine Learning
@krhoyt