



# runlinc Online Project 5: AI Object Recognition Library (runlinc Online Version)

---

## Contents

Introduction .....	1
Aim .....	1
Background .....	2
Write the Code.....	2
Run the Application .....	6
Extension.....	6

This project uses runlinc online which can be found here:

<https://www.runlinc.com/online/control.html>

(Note that HTTPS is required for this project)

Make sure to right-click the link and click 'open in new tab'.

Board IP:	<a href="https://runlinc.com">https://runlinc.com</a>
-----------	---

Check that your board IP says <https://runlinc.com> as shown to the right. If it does not try clearing your site data.

This project does not require a runlinc Wi-Fi Inventors' Kit.

**Please note:** FireFox Browser on Android Phone will allow you to select **front or rear** camera. Other browsers and devices ( such as Chrome, Safari, laptops and iPhone ) will open the **front** camera.

## Introduction

### Aim

This project will introduce you to how you can implement image recognition on runlinc. A demonstration of the project can be watched on the following links:

<https://youtu.be/odsZurCGUog>

[https://runlinc.com/Example/AI\\_Object\\_Voice\\_Spelling.html](https://runlinc.com/Example/AI_Object_Voice_Spelling.html)

## Background

Object recognition is one of the amazing applications of AI applied to camera or video. By real-time recording a video, we can recognize if an object on a camera is human or otherwise. This can help immensely for security reasons. Imagine the possibilities of the world with this technology. However, in this project, rather than coding object recognition from scratch, we will introduce you to a library that has already been made by others that we can use for object recognition.

The Library: MobileNets: small, low-latency, low-power models parameterized to meet the resource constraints of a variety of implementation cases. They can be built upon for classification, detection, embeddings and segmentation similar to how other popular large-scale models, such as Inception, are used. MobileNets trades off between latency, size and accuracy while comparing favorably with popular models from the literature.

This TensorFlow model does not require you to know about machine learning. It can take as input any browser-based image elements (<img>, <video>, <canvas> elements, for example) and returns an array of most likely predictions and their confidences.

## Write the Code

We'll add some style to the website with the following CSS:

```
html,
  body {
    overflow-x: hidden;
    background-color: #F5F5F5;
  }
.startButton {
  font-size: 12px;
  text-align: center;
  margin: 1%;
}
h1 {
  font-size: 40px;
  font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
```

```
}  
p {  
  font-size: 20px;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

Enter the following HTML:

```
<body cz-shortcut-listen="true">  
  <div class="startButton">  
    <button style="font-size:40px;background-color:#90ee90;border-radius:0.5rem;"  
    id="startButtonID" onclick="buttonChangeText();">Start</button>  
  </div>  
<head>  
  <meta charset="UTF-8">  
  <title>Webcam Object Classification</title>  
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.9.1/p5.min.js"></script>  
  <script src="https://unpkg.com/ml5@latest/dist/ml5.min.js" type="text/javascript"></script>  
</head>  
<body>  
  <h1>Webcam Object Classification</h1>  
</body>
```

Enter the following JavaScript:

```
function buttonChangeText(){  
  var initialMessage = "Initializing Setup";  
  var initialSpeakSetup = new SpeechSynthesisUtterance(initialMessage);  
  window.speechSynthesis.speak(initialSpeakSetup);  
};  
var classifier; var video; var resultsP; var msg1; var prevmsg;
```

```
function setup() {
  noCanvas();
  video = createCapture(VIDEO);
  classifier = ml5.imageClassifier('MobileNet', video, modelReady);
  resultsP = createP('Loading model and video...');
}
function modelReady() {
  console.log('Model Ready');
  classifyVideo();
}
function classifyVideo() {
  classifier.classify(gotResult);
}
async function gotResult(err, results) {
  resultsP.html(results[0].label + ' ' + nf(results[0].confidence, 0, 2));
  if(nf(results[0].confidence, 0, 2) > 0.35){
    if(results[0].label != prevmsg){
      msg1 = new SpeechSynthesisUtterance(results[0].label);
      window.speechSynthesis.speak(msg1);
      prevmsg = results[0].label;
    }
  }
  setTimeout(classifyVideo, 5000);
}
```

Expected runlinc control page:

Local

runlinc.com

Board IP

STEMSEL ↕

PORT	CONFIGURATION	NAME	STATUS
D2	DISABLED	<input style="width: 95%;" type="text"/>	
D4	DISABLED	<input style="width: 95%;" type="text"/>	
D5	DISABLED	<input style="width: 95%;" type="text"/>	
D12	DISABLED	<input style="width: 95%;" type="text"/>	
D13	DISABLED	<input style="width: 95%;" type="text"/>	
D14	DISABLED	<input style="width: 95%;" type="text"/>	
D15	DISABLED	<input style="width: 95%;" type="text"/>	
RX2	DISABLED	<input style="width: 95%;" type="text"/>	
TX2	DISABLED	<input style="width: 95%;" type="text"/>	
D18	DISABLED	<input style="width: 95%;" type="text"/>	
D19	DISABLED	<input style="width: 95%;" type="text"/>	
D21	DISABLED	<input style="width: 95%;" type="text"/>	
D22	DISABLED	<input style="width: 95%;" type="text"/>	
D23	DISABLED	<input style="width: 95%;" type="text"/>	
D25	DISABLED	<input style="width: 95%;" type="text"/>	
D26	DISABLED	<input style="width: 95%;" type="text"/>	
D27	DISABLED	<input style="width: 95%;" type="text"/>	
D32	DISABLED	<input style="width: 95%;" type="text"/>	
D33	DISABLED	<input style="width: 95%;" type="text"/>	
D34	DISABLED	<input style="width: 95%;" type="text"/>	
D35	DISABLED	<input style="width: 95%;" type="text"/>	
VP	DISABLED	<input style="width: 95%;" type="text"/>	
RNG1	DISABLED	<input style="width: 95%;" type="text"/>	

**CSS**

```
html,
body {
  overflow-x: hidden;
  background-color: #F0F0F0;
}
.startButton {
  font-size: 12px;
  text-align: center;
  margin: 10px;
}
h1 {
  font-size: 40px;
  font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
}
p {
  font-size: 20px;
  font-family: Arial, Helvetica, sans-serif;
}
```

**HTML**

```
<body cz-shortcut-listen="true">
<div class="startButton">
<button style="font-size:40px;background-color:#90ee90;border-radius:0.5rem;"
id="startButtonID" onclick="buttonChangeText();">Start</button>
</div>
<head>
<meta charset="UTF-8">
<title>Webcam Object Classification</title>
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.9.1/p5.min.js"></script>
<script src="https://unpkg.com/ml5@latest/dist/ml5.min.js" type="text/javascript"></script>
</head>
<body>
<h1>Webcam Object Classification</h1>
</body>
```

**JavaScript** Select Macro ↕ | select a device ↕ | Add Macro

```
function buttonChangeText() {
  var initialMessage = "Initializing Setup";
  var initialSpeakSetup = new SpeechSynthesisUtterance(initialMessage);
  window.speechSynthesis.speak(initialSpeakSetup);
};
var classifier; var video; var resultsP; var msg1; var prevmsg;

function setup() {
  noCanvas();
  video = createCapture(VIDEO);
  classifier = ml5.imageClassifier('MobileNet', video, modelReady);
  resultsP = createP('Loading model and video...');
}
function modelReady() {
  console.log('Model Ready');
  classifyVideo();
}
function classifyVideo() {
  classifier.classify(gotResult);
}
async function gotResult(err, results) {
  resultsP.html(results[0].label + " " + nf(results[0].confidence, 0, 2));
  if(nf(results[0].confidence, 0, 2) > 0.35){
    if(results[0].label != prevmsg){
      msg1 = new SpeechSynthesisUtterance(results[0].label);
      window.speechSynthesis.speak(msg1);
      prevmsg = results[0].label;
    }
  }
  setTimeout(classifyVideo, 5000);
}
```

**JavaScript Loop** Select Macro ↕ | select a device ↕ | Add Macro

Figure 1 Expected runlinc control page.

## Run the Application

Once you have written the code you can click 'Run Code' to start the program. You should see a page similar to the one shown in Figure 2, to the right.

You can now test out the object recognition with different things. The AI Voice will speak when the confidence level is above 0.35. Note, the program works best with an object in front of a black or white background, so the AI can easily see it without other distractions.

### Webcam Object Classification



ballpoint, ballpoint pen, ballpen, Biro 0.18

Figure 2 Expected website result.

## Extension

At the moment the AI Object Recognition code will text the name of the object under the video and a confidence number. When the confidence number is above 0.35 then it speaks what the object is. This extension adds to the speaking so for example if it says "Potato" then it will add to the speech: P O T A T O ( spell the word ). This can then be used as an educational app to learn English spelling.

In order to add the voice text into the above program, Find "window.speechSynthesis.speak(msg1);" in **JavaScript**, add this code after the located command:

```
content = results[0].label.split(',');
Text = content[0].toLowerCase();
letters = Text.split("");
letters1 = new SpeechSynthesisUtterance(letters);
window.speechSynthesis.speak(letters1);
```