



# runlinc AI Project 3: Simple Image Recognition Demonstration (E32W Version)

---

## Contents

Introduction .....	1
Part A: Design the Circuit on runlinc .....	2
Part B: Build the Circuit .....	3
Part C: Program the Circuit .....	4
Part D: Run the application .....	6

## Introduction

### Aim

Upload image and use third party machine learning JavaScript to compute the classification of the image with runlinc

### Background and Plan

Image processing with artificial intelligence (AI) is relatively simple. It thinks like a human brain.

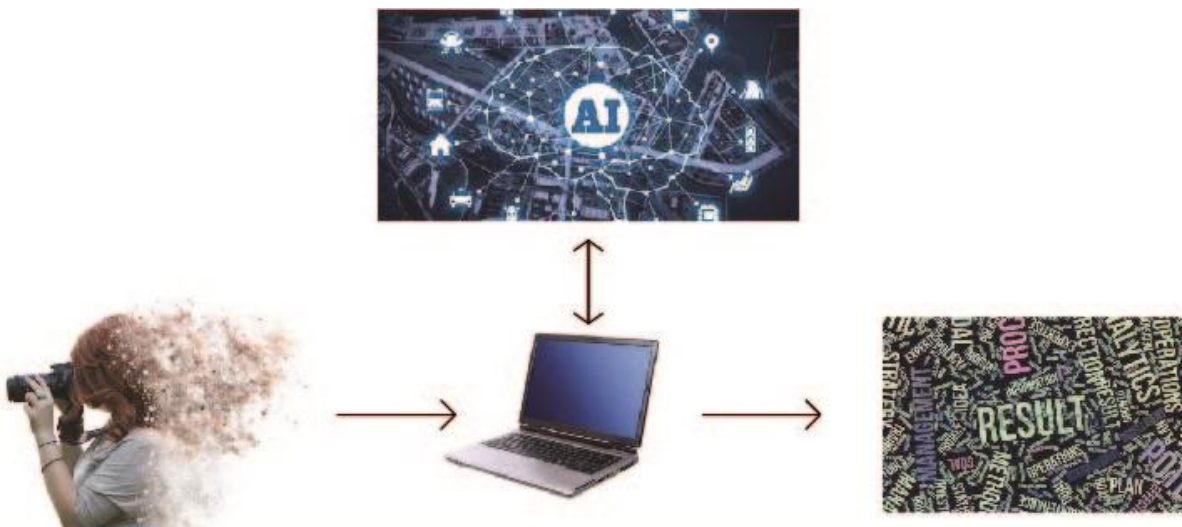


Figure 1 Image loaded into computer and through AI, result is shown at the end

Since we use the data base from third-party resource, we don't need to insert data by ourselves. We only need to know some syntaxes from that third-party resource for JavaScript. We won't be needing any inputs for the Microchip. Three LED will be used to signify that the respective process associated with the LED has been successful.

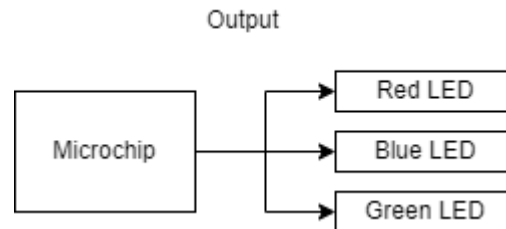


Figure 2 Block diagram of Microchip Outputs

## Runlinc Background

Runlinc is a web page inside a Wi-Fi chip. The programming is done inside the browsers compare to programming inside a chip. The runlinc web page inside the Wi-Fi chip will command the microchips to do sensing, control, data logging Internet of Things (IoT). It can predict and command.

## Part A: Design the Circuit on runlinc

**Note:** refer to runlinc Wi-Fi setup guide document to connect to runlinc

D18	DIGITAL_OUT	blue	OFF
D19	DIGITAL_OUT	red	OFF
D21	DIGITAL_OUT	green	OFF

Figure 3 Digital outputs and description

For port D18 name it blue and set it as DIGITAL\_OUT.

For port D19 name it red set it as DIGITAL\_OUT.

For port D21 name it green set it as DIGITAL\_OUT.

## Part B: Build the Circuit

Use the STEMSEL E32 board to connect the hardware. For this project we are using both the left and right I/O ports, with **negative port (-ve)** on the outer side, **positive port (+ve)** on the middle and **signal port (s)** on the inner side (as shown below).

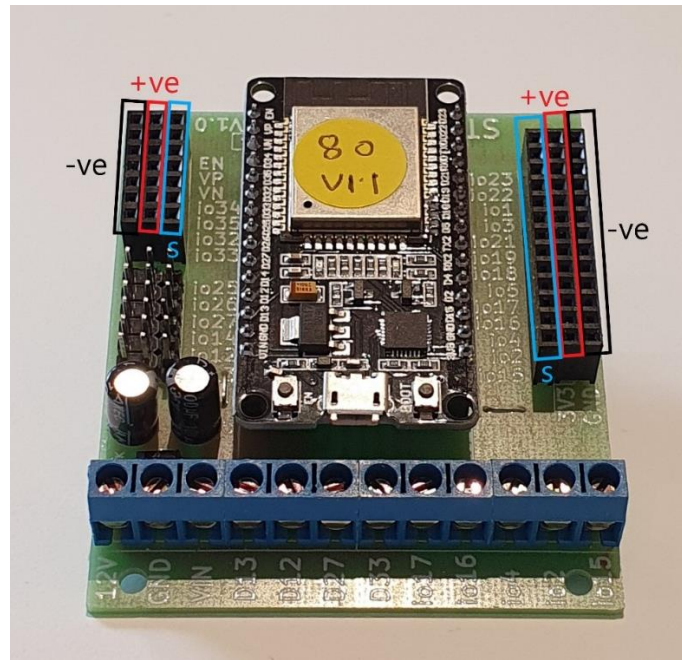


Figure 4: Setup on E32W board

There is only one I/O parts we are using for this project, a 4-pin RGB LED, its respective pins are shown in the figure below.

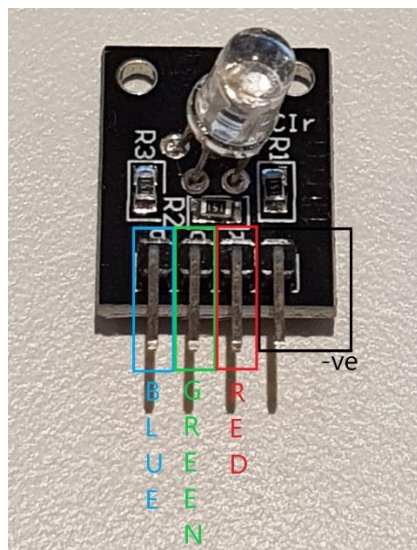


Figure 5: I/O parts with negative, positive, and signal pins indicated

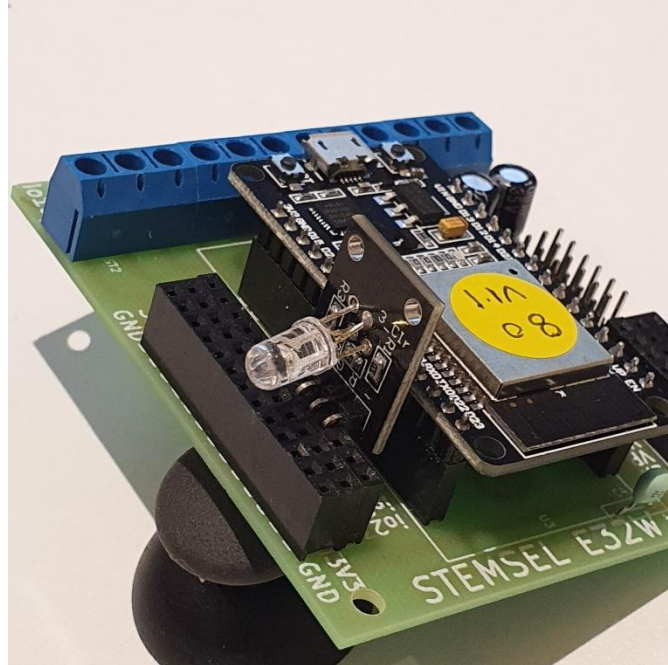


Figure 6: Setup on E32W board

## **Wiring Instructions**

- Connect white wires of each respective LED to their respective pin.
- All part's pin need plug in **signal port (s)**
  - a. C18->DIGITAL\_OUT : Green
  - b. C19->DIGITAL\_OUT : Red
  - c. C21->DIGITAL\_OUT : Blue

## **Part C: Program the Circuit**

As the main purpose of this project is to demonstrate the image recognition capability of the microchip and coding, you'll be mainly just copy and pasting the code to see the effects of the code.

### **HTML:**

```
<script src="https://unpkg.com/ml5@0.1.3/dist/ml5.min.js"></script>
<h1>Image classification</h1>
<input id="load">
<input type="submit" onclick="imgProcess()">
<p>This labeled as:
  <span id="result">...</span> <br>with a confidence of
  <span id="probability">...</span></p>
<img src="" crossorigin="anonymous" id="image" width="400">
<p style="font-size:10px">supported by MobileNet</p>
```

### Javascript:

```
var type = "";
turnOn(blue);
function imgProcess() {
  // The image we want to classify
  var loadIMG = document.getElementById("load");
  var image = document.getElementById('image');
  image.setAttribute("src", loadIMG.value);
  // The result tag in the HTML
  const result = document.getElementById('result');
  // The probability tag in the HTML
  const probability = document.getElementById('probability');

  // Initialize the Image Classifier method with MobileNet
  const classifier = ml5.imageClassifier('MobileNet', function () {
    console.log('Model Loaded!');
  });

  // Make a prediction with the selected image
  // This will return an array with a default of 10 options with their probabilities
  classifier.predict(image, async function (err, results) {
    result.innerText = results[0].className;
    type = results[0].className;
    probability.innerText = results[0].probability.toFixed(4);
    var utterance = new SpeechSynthesisUtterance(results[0].className);
    speechSynthesis.speak(utterance);
    switch (type) {
      case 'tiger shark, Galeocerdo cuvieri': //example:
        https://www.dw.com/image/45601658_401.jpg
        turnOn(red);
        turnOff(blue);
        turnOff(green);
        break;

      case 'matchstick':
        turnOn(green);
        turnOff(red);
        turnOff(blue);
        break;

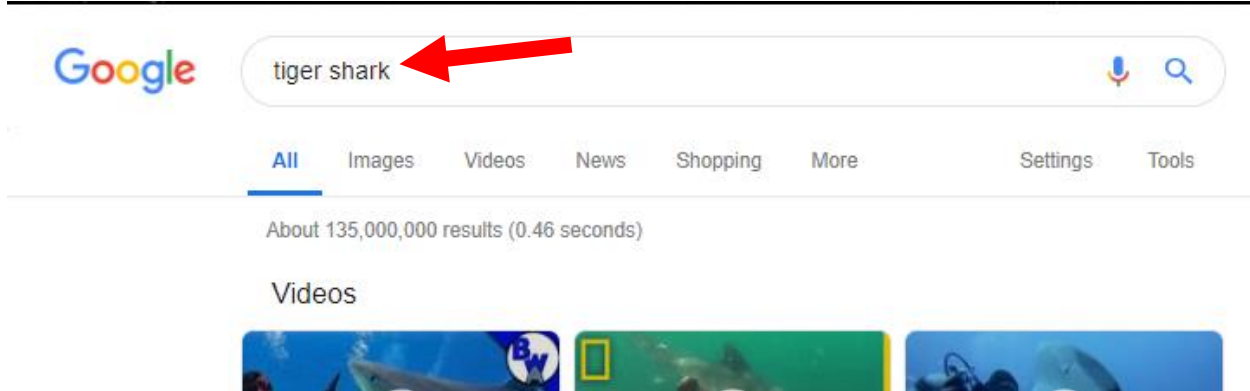
      case value:
        turnOff(blue);
        turnOff(red);
        turnOff(green);
        break;

      default:
        break;
    }
  });
}
```

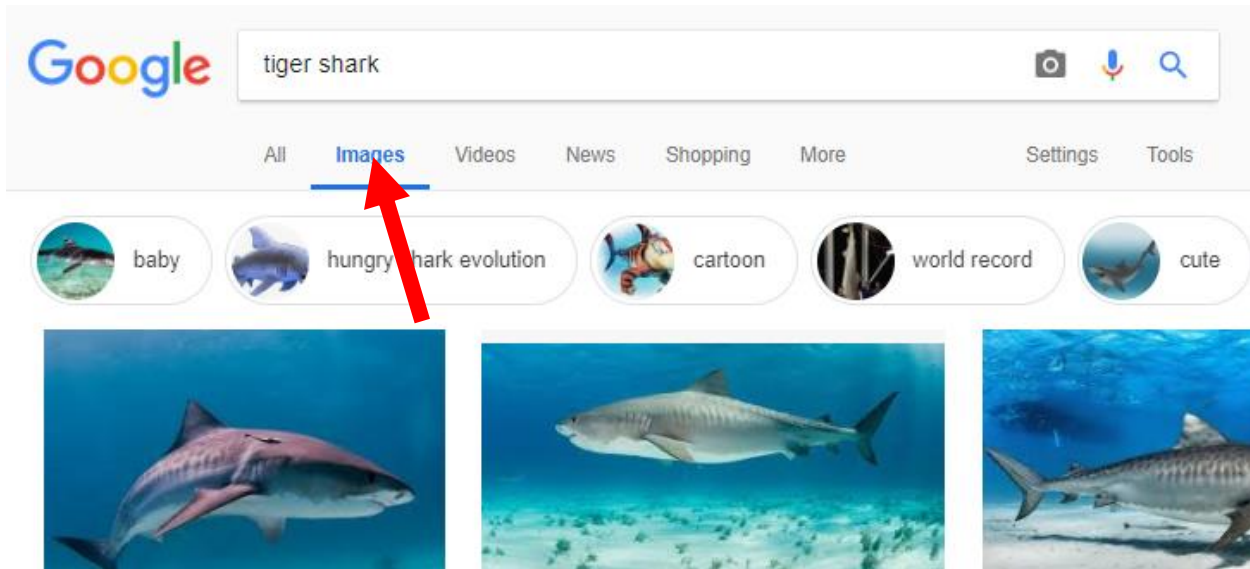
## Part D: Run the application

### How to get image link:

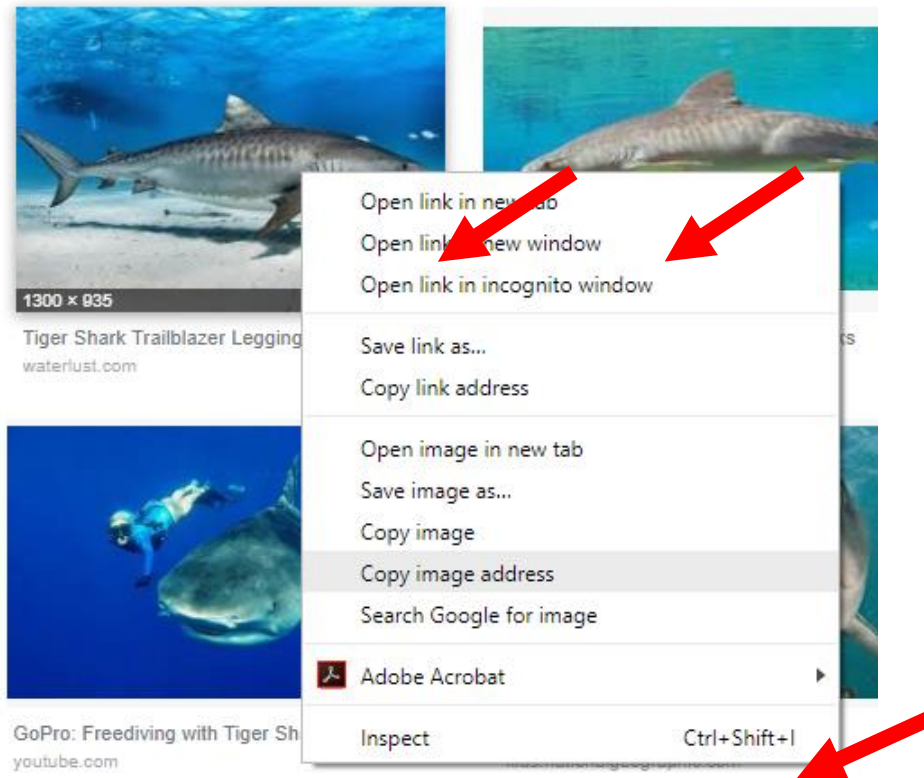
Step 1: Search any image you like. Example: tiger shark



Step 2: Click on Image



Step 3: Right click on the image you like and select “Copy image address”



Step 4: Paste it in runlinc page and click submit

## Image classification

This labeled as: tiger shark, *Galeocerdo cuvieri*  
with a confidence of 0.9988



supported by MobileNet