runlinc AI Project 5:
AI Object Recognition Library (STEMSEL Version)

Contents

Introduction

Aim

This project will introduce you with how to implement image recognition on runlinc. A demonstration of the project can be watched on the following link: https://www.youtube.com/watch?v=1oR-HeUsfUQ

Background

Object recognition is one of the amazing applications of Image recognition applied to camera/video. By real-time recording a video, we can recognise if an object in 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 the library that has already been made by others that we can use for object recognition.

The Library: MobileNets: small, low-latency, low-power models parameterised 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 favourably 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.

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: The Plan and Circuit on runlinc

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

For this project, we won't be using any input or output (I/O).

However, the webpage will use your computer's connected camera. For example, if you used a laptop, it will use your laptop's inbuilt camera. If you use a desktop, it will connect to the connected camera or if not connected, will not show anything.

Also, if the recognition is above 0.5 ratios, it will produce speech audio of the result.

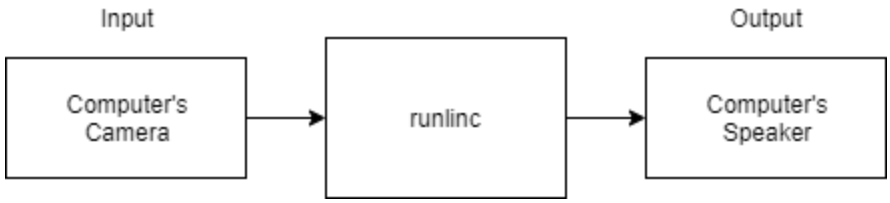

Figure 1 runlinc I/O on the host computer

Part B: Preparation

You will need to set up a pagekite connection. Please go to the pagekite setup document to learn more about setting up a pagekite connection.

Part C: Program the Circuit

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

```
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:

```
<head>
  <meta charset="UTF-8">
  <title>Webcam Object Classification</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/p5.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.9.0/addons/p5.dom.min.js"></script>
  <script src="https://unpkg.com/ml5@0.4.3/dist/ml5.min.js" type="text/javascript"></script>
</head>
<body>
  <h1>Webcam Object Classification</h1>
</body>
```

Enter the following JavaScript:

```
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) > 0.5){
    if(results[0].label != prevmsg){
      msg1 = new SpeechSynthesisUtterance(results[0].label);
      window.speechSynthesis.speak(msg1);
      prevmsg = results[0].label;
    }
  }
  await mSec( 2500 );
  classifyVideo();
}
```

Expected runlinc control page:

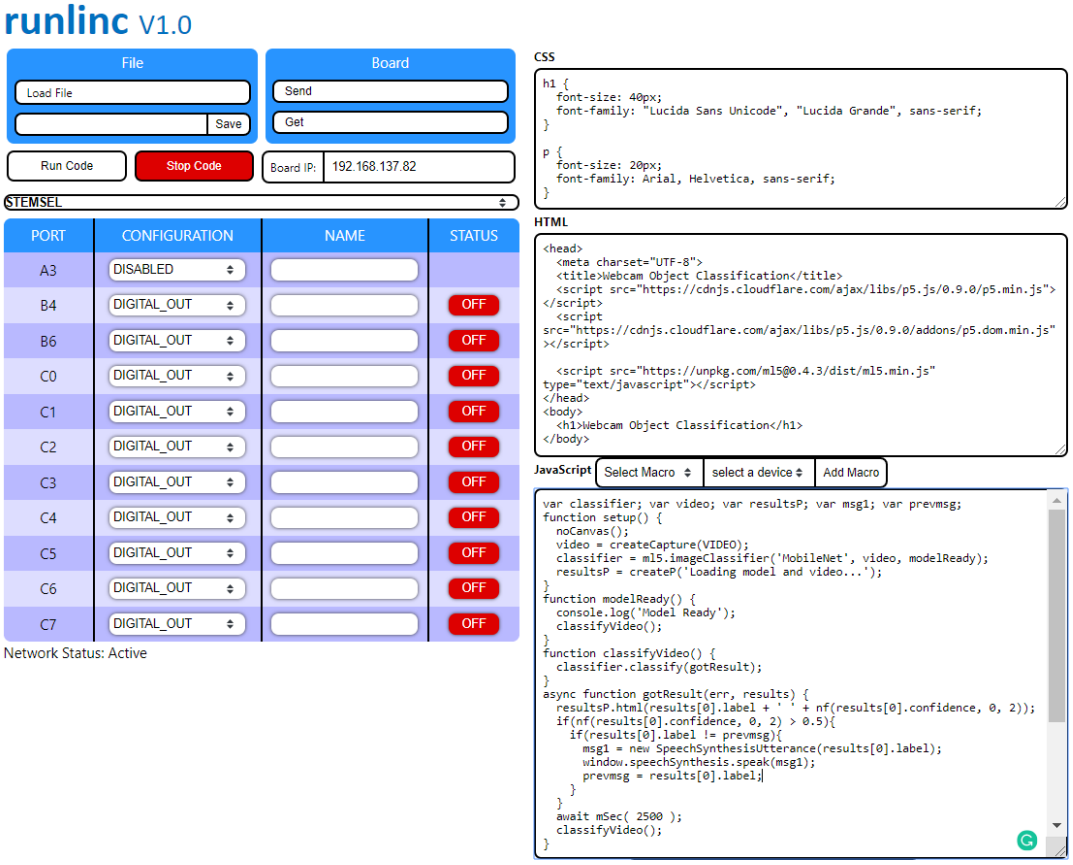

Figure 2 Expected runlinc control page.

Part D: Run the Application

When you finish implementing the code to the STEMSEL board, remember to send the code to the board***. You then SHOULD connect to the webpage using your pagekite link using https prefix (i.e. https://***.pagekite.me where *** is your pagekite).*** Once the page is connected, you should have the following page in Figure 3.

Then you can play around object recognition with many kinds of stuff!

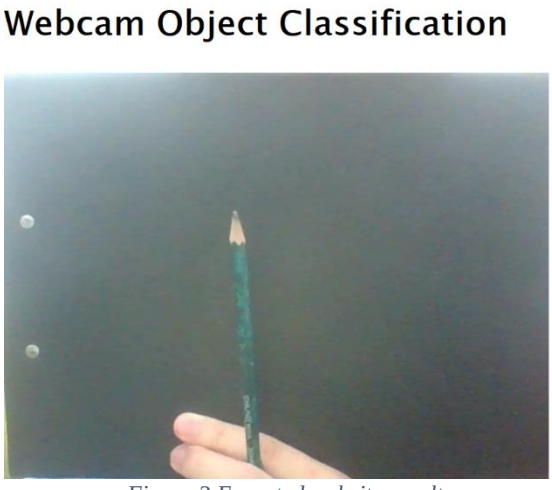

Figure 3 Expected website result.

Appendix A: Fail to finish loading

Some of you might find that the code will not work as the page is stuck on loading sign.

Then it is the most likely problem that you are using the latest browser with a security patch. Your runlinc is running on http connection whereas the library is using an https connection. Latest browser security patch does not allow the http and https resources to mix.

Therefore, the JavaScript part needs to be modified for the project to work, and this modification works only on Firefox (as of now).

The new JavaScript will become:

```
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) > 0.5){
    if(results[0].label != prevmsg){
      msg1 = new SpeechSynthesisUtterance(results[0].label);
      window.speechSynthesis.speak(msg1);
      prevmsg = results[0].label;
    }
  }
  await mSec( 2500 );
  classifyVideo();
}

function mSec(delay){
  return (
    new Promise(
      (resolve) => setTimeout(
        () => resolve(true),
        delay
      )
    )
  );
};
```