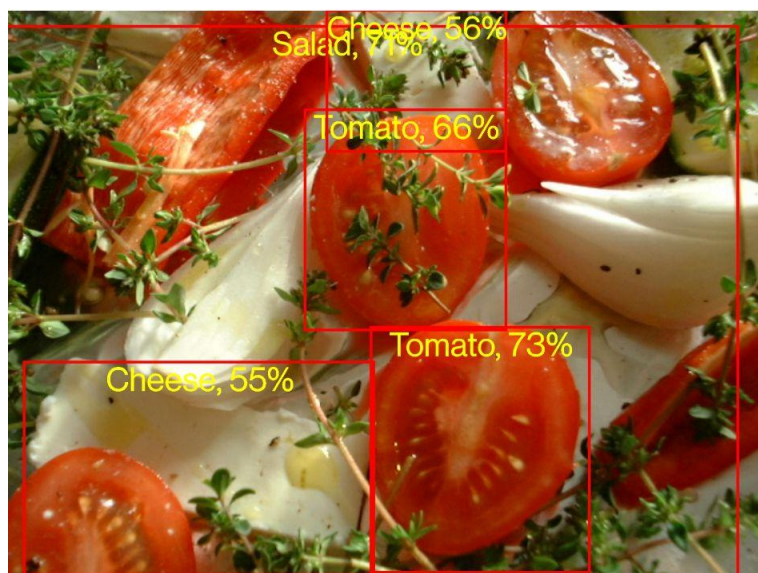
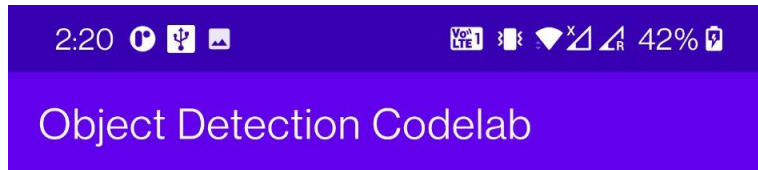


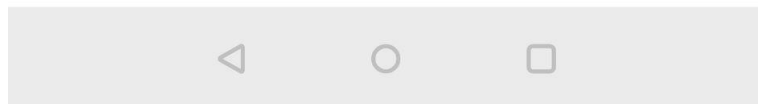
Device ML skills

Part A - Mobile App and Image training on device: Build and deploy a custom object detection model using tf lite on android simulator

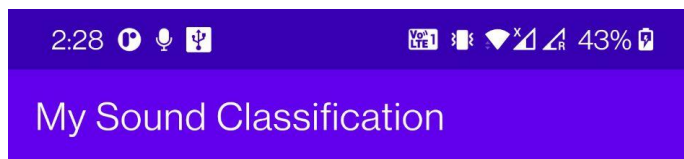
Screenshots - Part A



Select a preset image or take a new photo



Screenshots - Part B



Recording

Red Crossbill -> 0.44998854

Number Of Channels: 1
Sample Rate: 16000



2:28



Von
LTE 1



43%

My Sound Classification

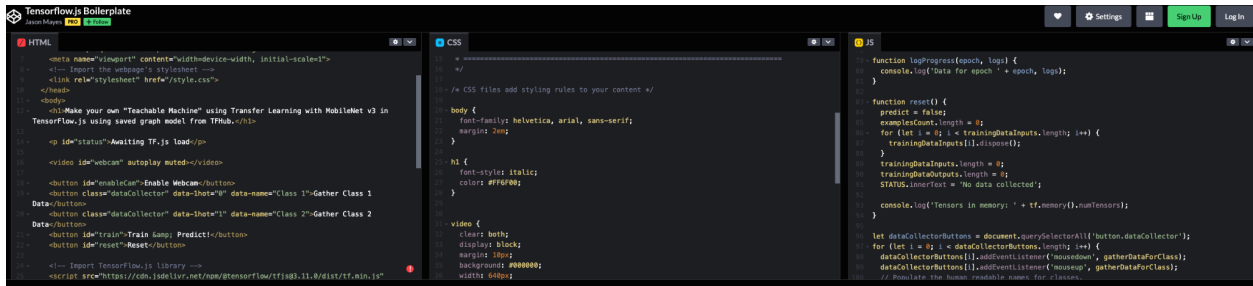
Recording

White-breasted Wood-Wren ->
0.6844293

Number Of Channels: 1
Sample Rate: 16000



Screenshots - Part C



```
HTML
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Import the webpage's stylesheet -->
<link rel="stylesheet" href="/style.css">
</head>
<body>
  <!-- Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub. -->
  <p id="status">Awaiting TF.js load</p>
  <video id="webcam" autoplay muted</video>
  <button id="enableCam">Enable Webcam</button>
  <button class="dataCollector" data-hot="0" data-name="Class 1">Gather Class 1 Data</button>
  <button class="dataCollector" data-hot="1" data-name="Class 2">Gather Class 2 Data</button>
  <button id="train">Train & Predict</button>
  <button id="reset">Reset</button>
  <!-- Import TensorFlow.js library -->
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@3.11.0/dist/tf.min.js">

```

```
CSS
/*
 * CSS files add styling rules to your content */
/*
 */
body {
  font-family: helvetica, arial, sans-serif;
  margin: 20px;
}
h1 {
  font-style: italic;
  color: #FF69B4;
}
video {
  clear: both;
  display: block;
  margin: 10px;
  background: #000000;
  width: 600px;
}

```

```
JS
function logProgress(epoch, logs) {
  console.log('Data for epoch ' + epoch, logs);
}

function reset() {
  predict = false;
  examplesCount.length = 0;
  for (let i = 0; i < trainingDataInputs.length; i++) {
    trainingDataInputs[i].dispose();
  }
  trainingDataInputs.length = 0;
  trainingDataOutputs.length = 0;
  STATUS.innerText = 'No data collected';
  console.log('Tensors in memory: ' + tf.memory().numTensors);
}

let dataCollectorButtons = document.querySelectorAll('button.dataCollector');
for (let i = 0; i < dataCollectorButtons.length; i++) {
  dataCollectorButtons[i].addEventListener('mousedown', gatherDataForClass);
  dataCollectorButtons[i].addEventListener('mouseup', gatherDataForClass);
}
// Populate the home-readable names for classes.

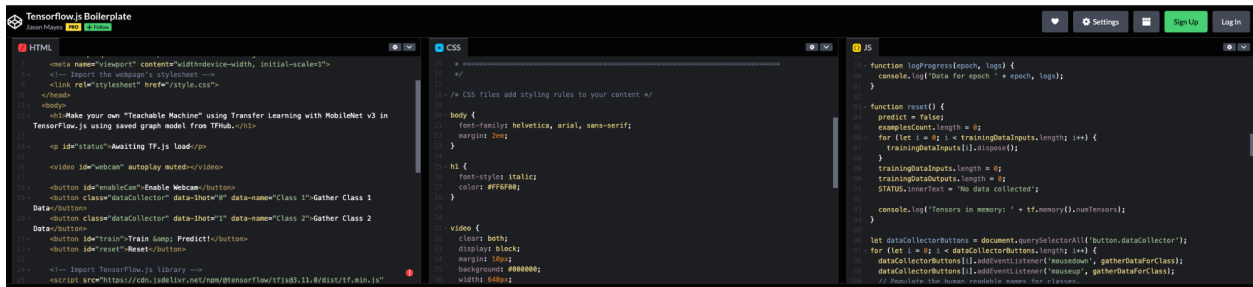
```

Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

Prediction: Class 2 with 99% confidence



Gather Class 1 Data Gather Class 2 Data Train & Predict Reset



```
HTML
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Import the webpage's stylesheet -->
<link rel="stylesheet" href="/style.css">
</head>
<body>
  <!-- Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub. -->
  <p id="status">Awaiting TF.js load</p>
  <video id="webcam" autoplay muted</video>
  <button id="enableCam">Enable Webcam</button>
  <button class="dataCollector" data-hot="0" data-name="Class 1">Gather Class 1 Data</button>
  <button class="dataCollector" data-hot="1" data-name="Class 2">Gather Class 2 Data</button>
  <button id="train">Train & Predict</button>
  <button id="reset">Reset</button>
  <!-- Import TensorFlow.js library -->
  <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@3.11.0/dist/tf.min.js">

```

```
CSS
/*
 * CSS files add styling rules to your content */
/*
 */
body {
  font-family: helvetica, arial, sans-serif;
  margin: 20px;
}
h1 {
  font-style: italic;
  color: #FF69B4;
}
video {
  clear: both;
  display: block;
  margin: 10px;
  background: #000000;
  width: 600px;
}

```

```
JS
function logProgress(epoch, logs) {
  console.log('Data for epoch ' + epoch, logs);
}

function reset() {
  predict = false;
  examplesCount.length = 0;
  for (let i = 0; i < trainingDataInputs.length; i++) {
    trainingDataInputs[i].dispose();
  }
  trainingDataInputs.length = 0;
  trainingDataOutputs.length = 0;
  STATUS.innerText = 'No data collected';
  console.log('Tensors in memory: ' + tf.memory().numTensors);
}

let dataCollectorButtons = document.querySelectorAll('button.dataCollector');
for (let i = 0; i < dataCollectorButtons.length; i++) {
  dataCollectorButtons[i].addEventListener('mousedown', gatherDataForClass);
  dataCollectorButtons[i].addEventListener('mouseup', gatherDataForClass);
}
// Populate the home-readable names for classes.

```

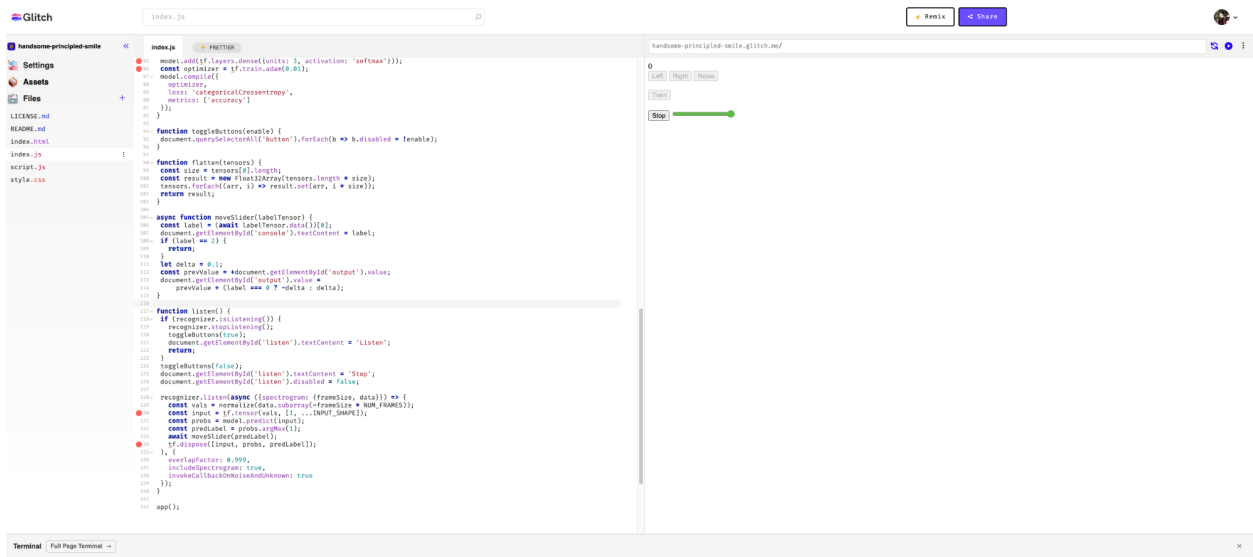
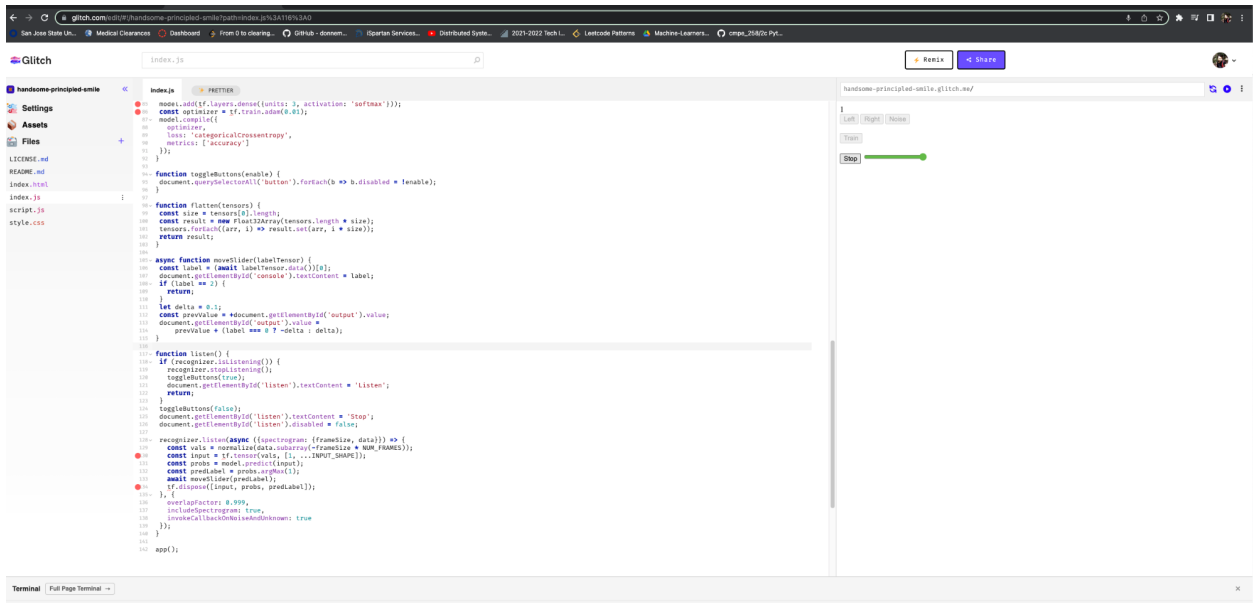
Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.

Prediction: Class 1 with 99% confidence



Gather Class 1 Data Gather Class 2 Data Train & Predict Reset

Screenshots - Part D



Screenshots - Part E

4:00

VoWiFi 3G+ 59%

ML Kit Codelab



Test Image 1 (Text)

FIND TEXT

FIND FACE CONTOUR

4:00



VoLTE



59%



59%



59%

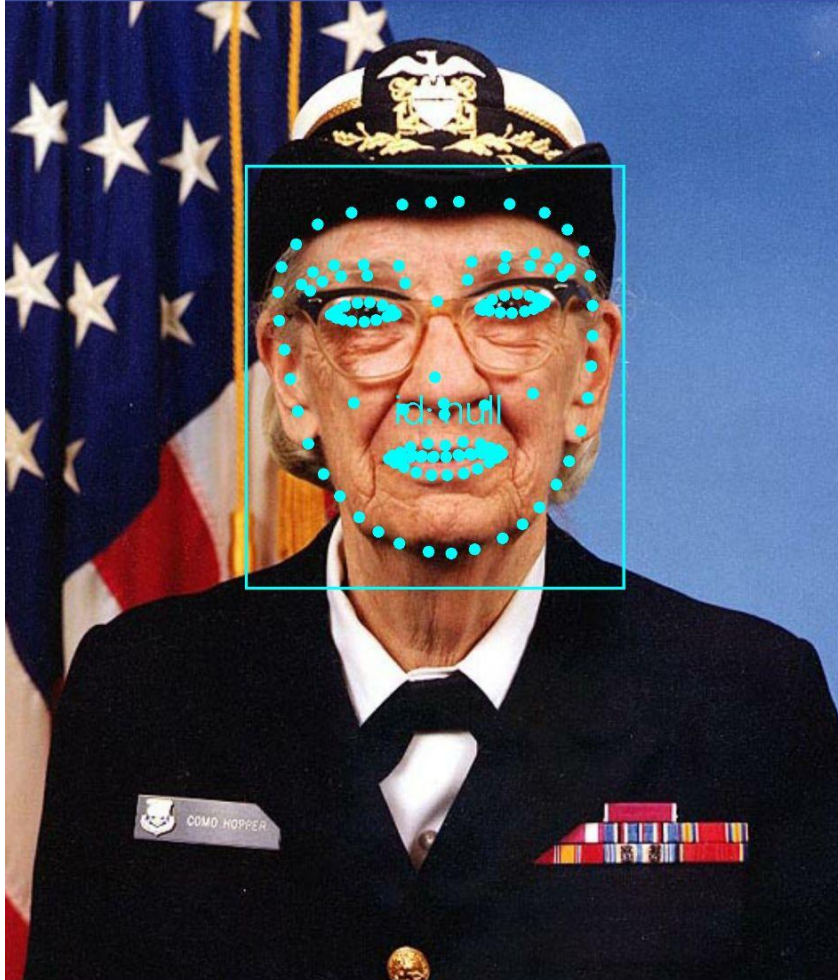


59%



59%

ML Kit Codelab



Test Image 2 (Face)



FIND TEXT

FIND FACE CONTOUR



4:00



ML Kit Codelab



Test Image 2 (Face)



FIND TEXT

FIND FACE CONTOUR



4:18



VoLTE



61%

An Easy & Proven Way to
Build Good Habits & Break Bad Ones

Center text in box

James Clear

ENGLISH

An Easy & Proven Way to
Build Good Habits & Break
Bad Ones

English



An Easy & Proven
Way to

powered by Google Translate