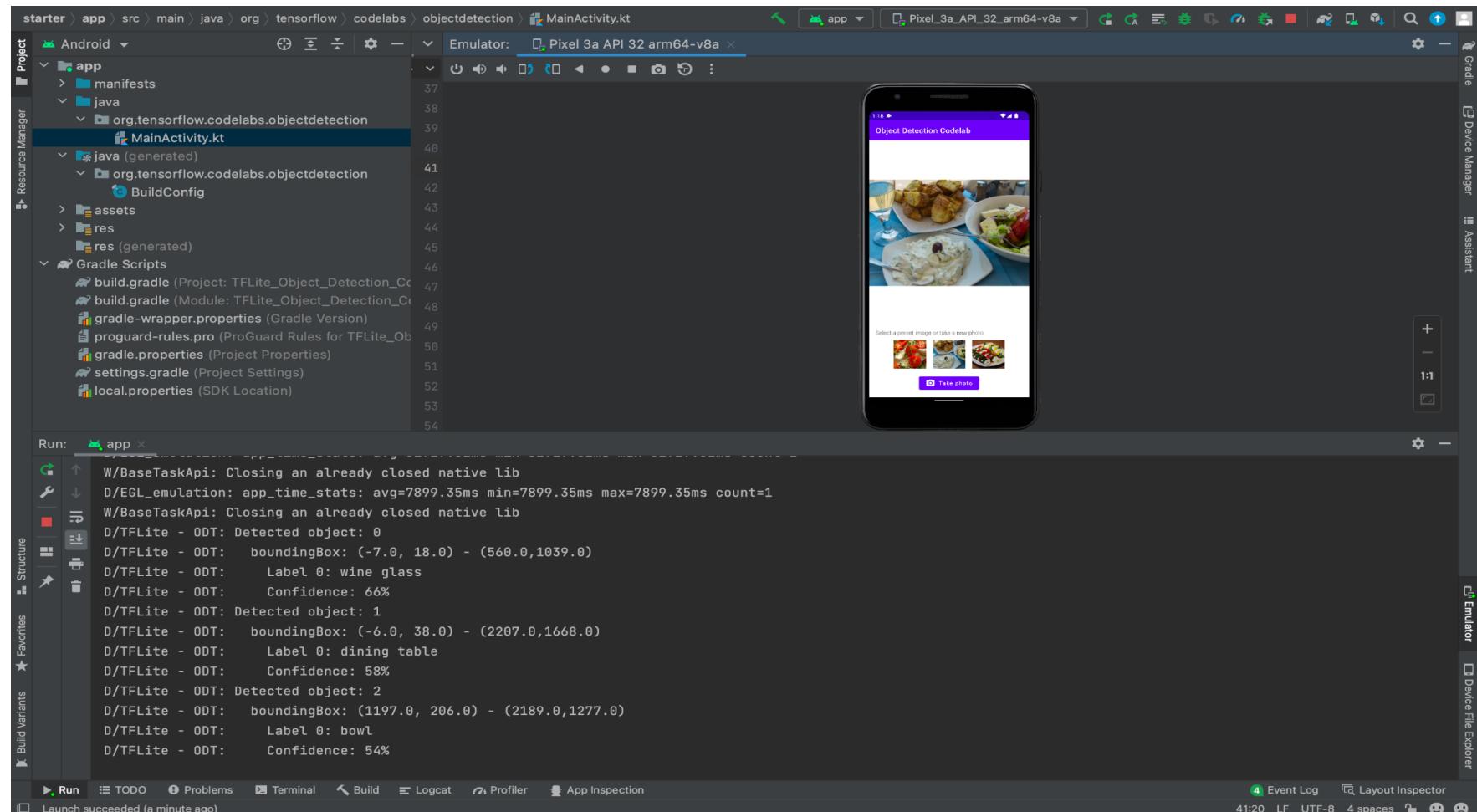


Ondevice ML skills

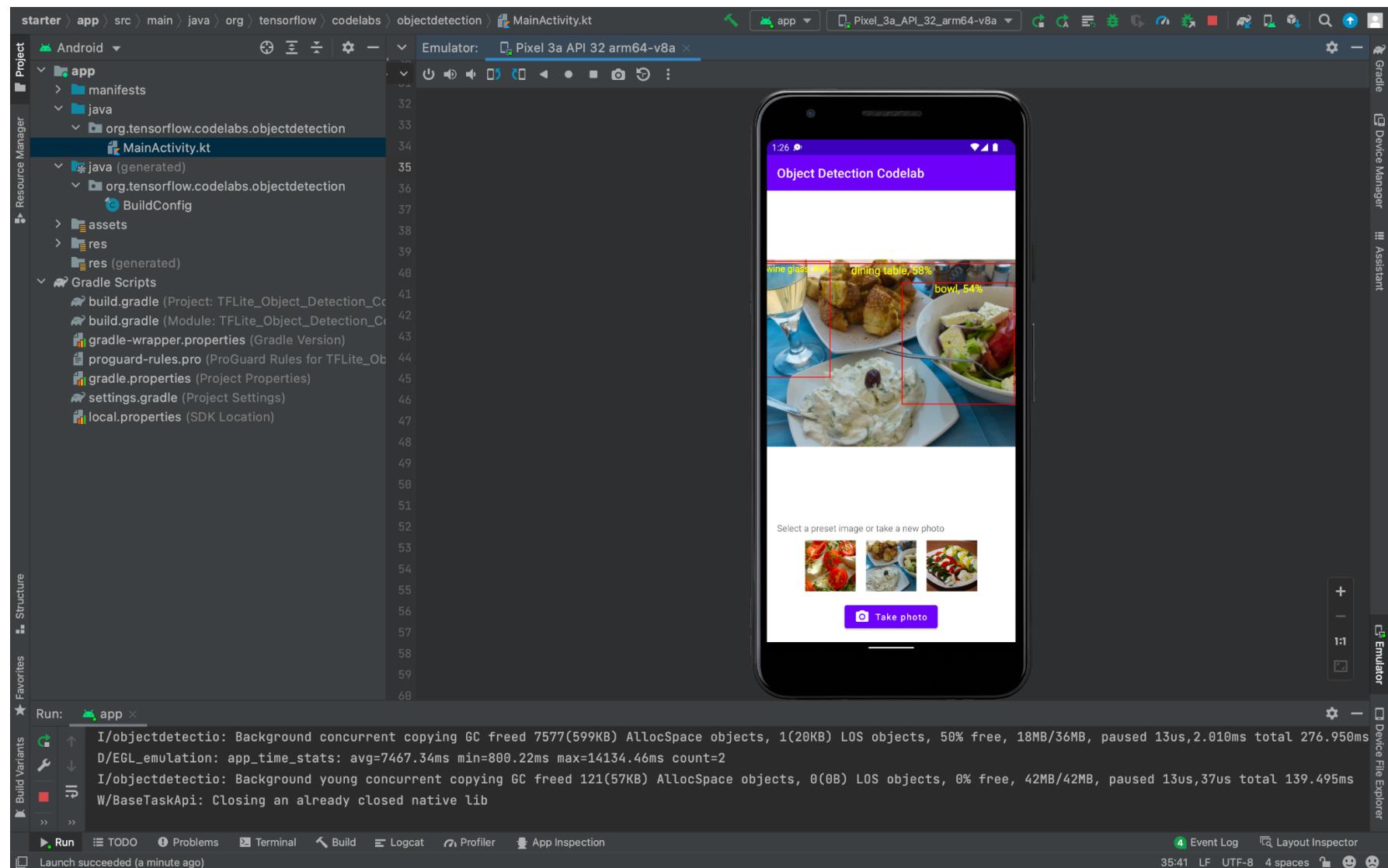
a) Mobile App and Image training on the device: Build and deploy a custom object detection model using tflite on the android simulator ([Tflite-Object-detection](#))

- Followed all the steps per the Codelab.

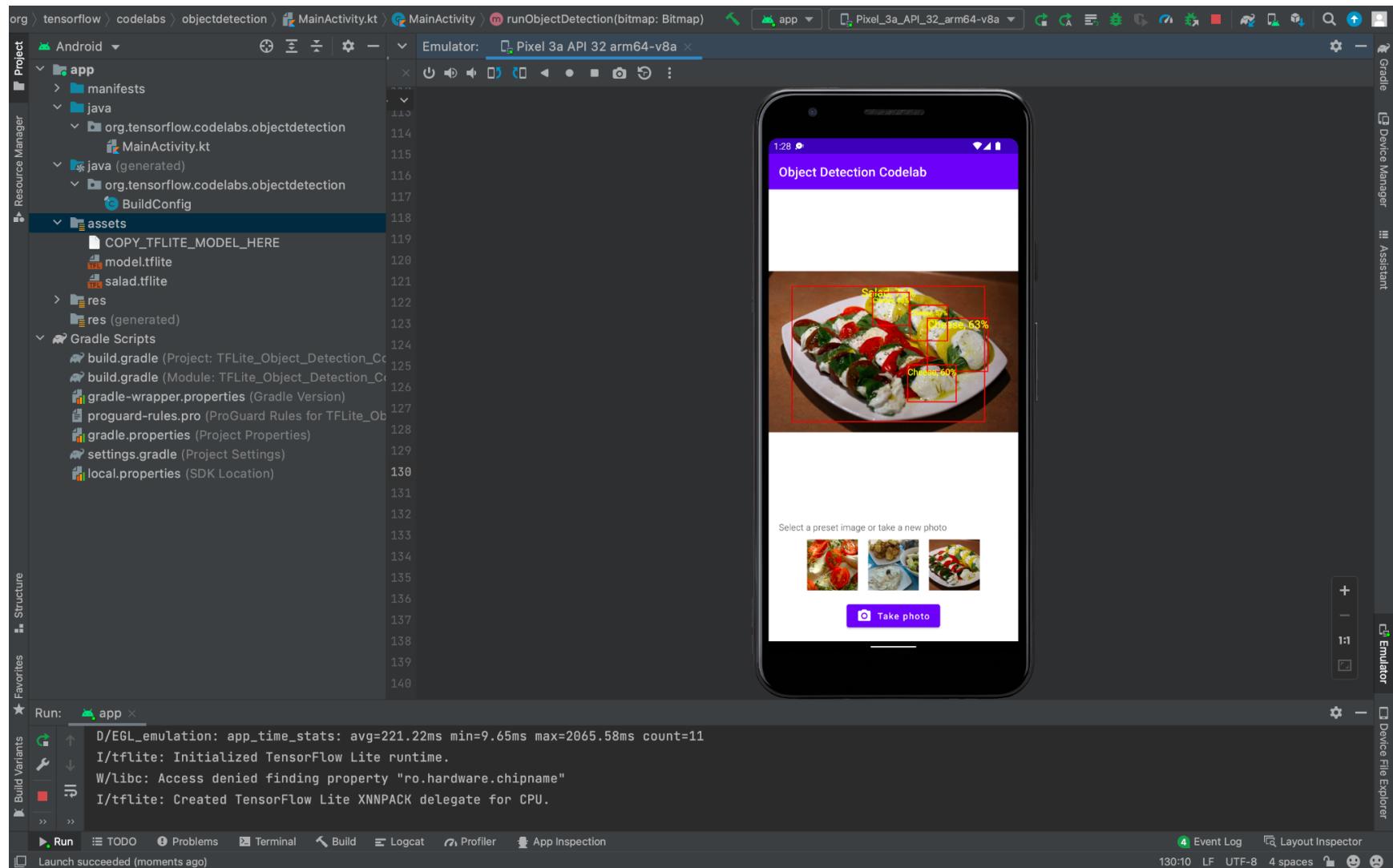
OBJECT DETECTION SCREENSHOT:



Ondevice ML skills



Ondevice ML skills

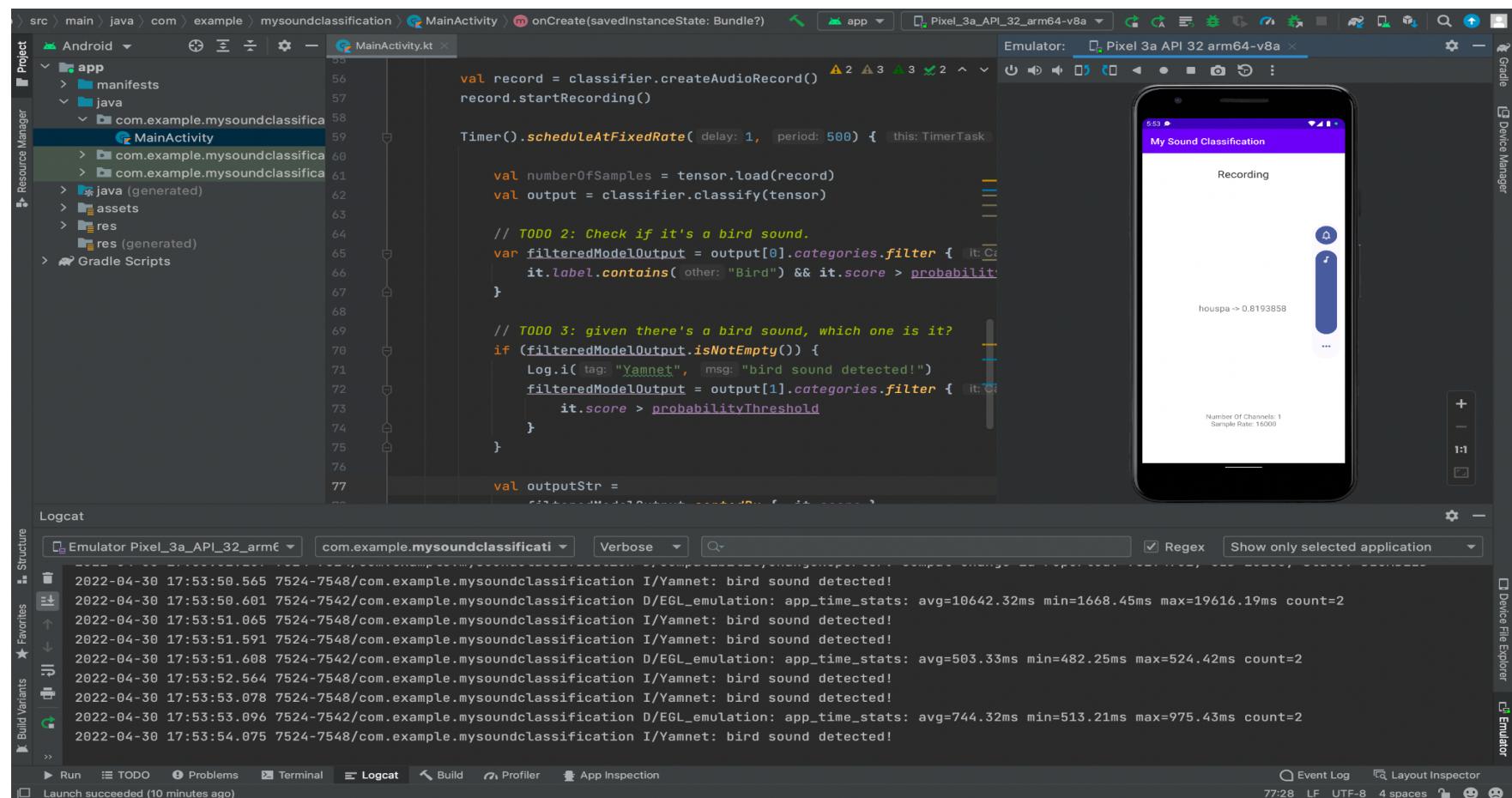


Ondevice ML skills

- b) Mobile app and audio training on the device: Build a custom pre-trained Audio Classification model ([Audio-classification\(Birds\)](#))

BIRD AUDIO CLASSIFICATION SCREENSHOT:

Downloaded the Model from the Codelab and used it in the Tflite and verified it with some random bird sounds. It worked well.



Ondevice ML skills

c) Web app and image training on the device: TensorFlow.js Transfer Learning Image Classifier ([Image Classification](#)) - Used Codepen to create & run the web app and tested the model performance.



The screenshot shows a CodePen editor window titled "Tensorflow.js Boilerplate". The URL is https://codepen.io/avigeth/pen/RwQwKMq. The editor interface includes tabs for HTML, CSS, and JS, and a preview area.

HTML:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Transfer Learning - TensorFlow.js</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

CSS:

```
body {
  font-family: helvetica, arial, sans-serif;
  margin: 2em;
}
```

JS:

```
const STATUS = document.getElementById('status');
const VIDEO = document.getElementById('webcam');
const ENABLE_CAM_BUTTON =
document.getElementById('enableCam');
const RESET_BUTTON = document.getElementById('reset');
const TRAIN_BUTTON = document.getElementById('train');
```

Prediction: Class 1 with 99% confidence



Below the video feed are four buttons: "Gather Class 1 Data", "Gather Class 2 Data", "Train & Predict!", and "Reset".

At the bottom of the CodePen interface, there are links for "Console", "Assets", "Comments", and "Keys". On the right, there are buttons for "Delete", "Add to Collection", "Fork", "Embed", "Export", and "Share". A status message "Last saved LESS THAN A MINUTE AGO" is also present.

Ondevice ML skills

Tensorflow.js Boilerplate

https://codepen.io/avigeth/pen/RwQwKMq

Tensorflow.js Boilerplate

Avinash

HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Transfer Learning - TensorFlow.js</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

CSS

```
body {
  font-family: helvetica, arial, sans-serif;
  margin: 2em;
}
```

JS

```
const STATUS = document.getElementById('status');
const VIDEO = document.getElementById('webcam');
const ENABLE_CAM_BUTTON =
document.getElementById('enableCam');
const RESET_BUTTON = document.getElementById('reset');
const TRAIN_BUTTON = document.getElementById('train');
```

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

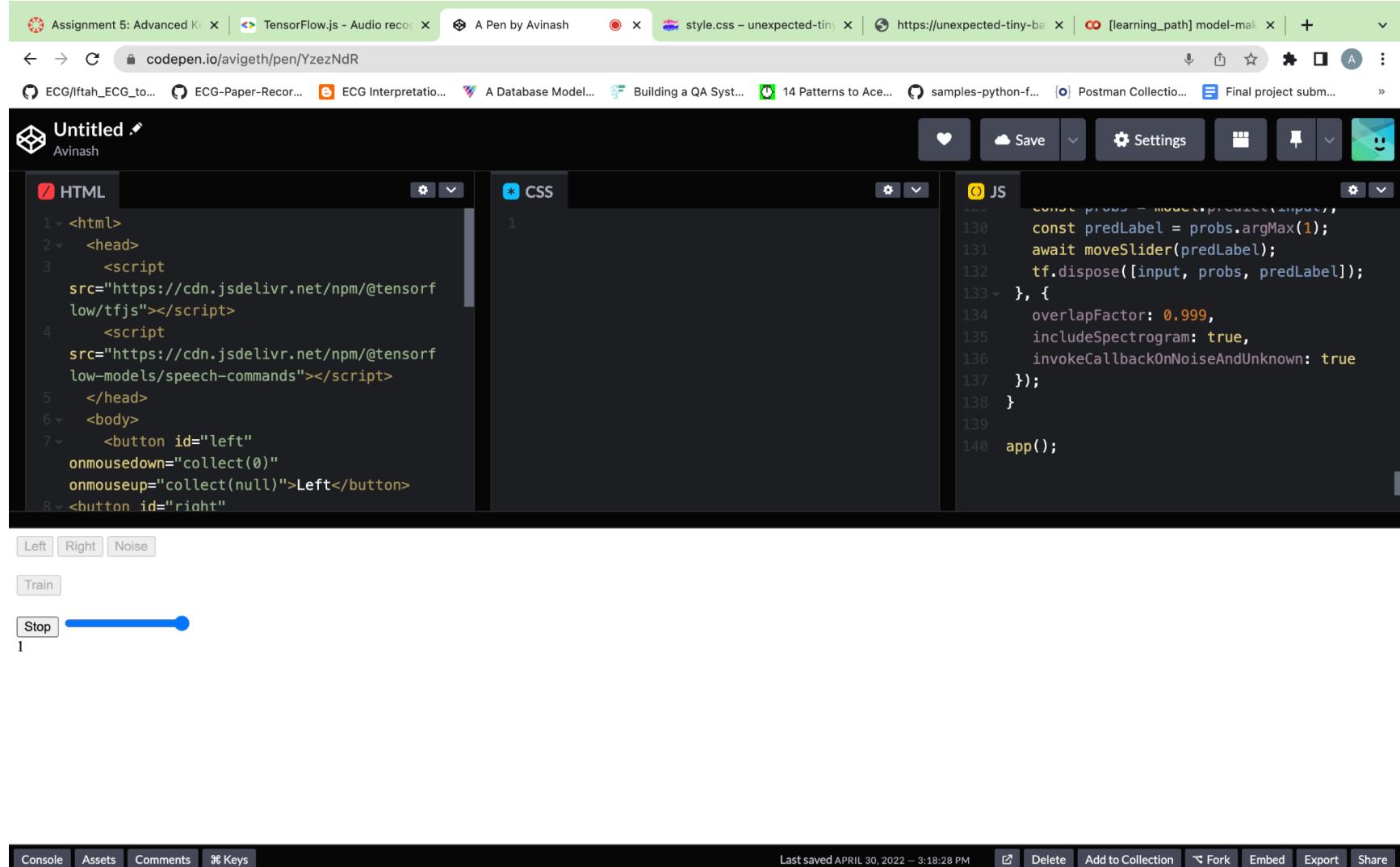


Gather Class 1 Data Gather Class 2 Data Train & Predict! Reset

Last saved LESS THAN A MINUTE AGO Delete Add to Collection Fork Embed Export Share

Ondevice ML skills

d) Webapp and audio training on the device: TensorFlow.js - Audio recognition using transfer learning ([Audio recognition](#)) - used codepen to develop an audio classifier web app and tested it. Was working as expected.



The screenshot shows a CodePen interface with the following details:

- Title:** Untitled
- Author:** Avinash
- HTML Tab:**

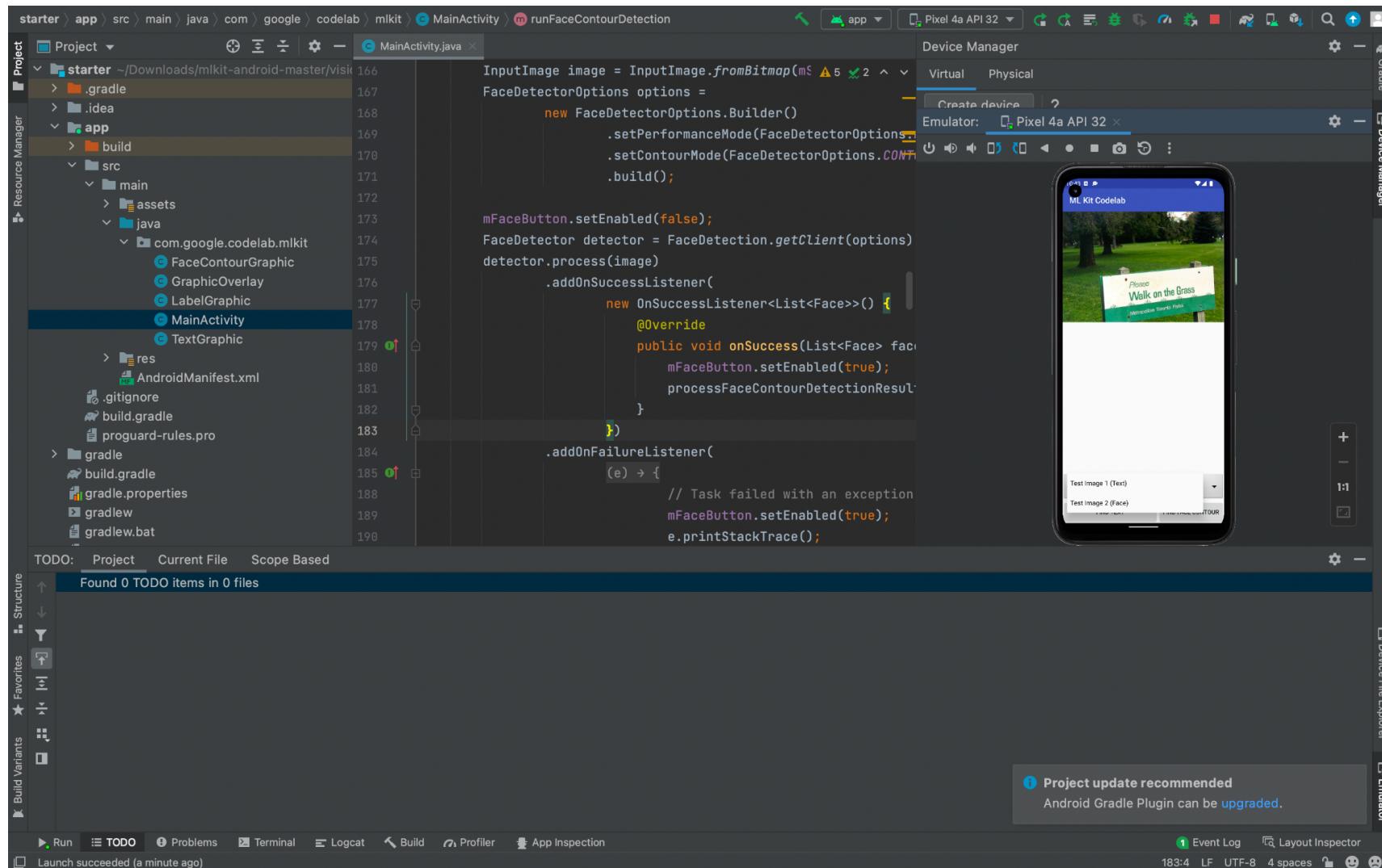
```
1 <html>
2   <head>
3     <script
4       src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
5     <script
6       src="https://cdn.jsdelivr.net/npm/@tensorflow-models/speechCommands"></script>
7   </head>
8   <body>
9     <button id="left"
10       onmousedown="collect(0)"
11       onmouseup="collect(null)">Left</button>
12     <button id="right"
13       onmousedown="collect(1)"
14       onmouseup="collect(null)">Right</button>
15   </body>
16 </html>
```
- CSS Tab:**

```
1
```
- JS Tab:**

```
130 const probs = model.predict(samples);
131 const predLabel = probs.argMax(1);
132 await moveSlider(predLabel);
133 tf.dispose([input, probs, predLabel]);
134 }, {
135   overlapFactor: 0.999,
136   includeSpectrogram: true,
137   invokeCallbackOnNoiseAndUnknown: true
138 }
139
140 app();
```
- Buttons at the bottom:** Left, Right, Noise, Train, Stop
- Slider:** A horizontal slider with a blue track and a black handle, currently set to 1.
- Page footer:** Console, Assets, Comments, ⌘ Keys, Last saved APRIL 30, 2022 – 3:18:28 PM, Delete, Add to Collection, Fork, Embed, Export, Share

Ondevice ML skills

- e) Using out-of-box SDK to do ML on-device: Use ML Kit to perform [Recognize text and facial features.](#), and [Recognize, Identify Language and Translate text with ML Kit and CameraX: Android](#)



Ondevice ML skills

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "mlkit" and contains an "app" module. The "app" module has Java files like `MainActivity.java` and `FaceContourGraphic.java`, and assets such as `grace_hopper.jpg` and `Please_walk_on_the_grass.jpg`.
- MainActivity.java:** The code implements face detection and contour drawing. It uses the `FaceDetectorOptions` builder to set performance mode to `PERF` and contour mode to `CONTOUR`. It adds an on-success listener to handle detected faces and an on-failure listener.
- Emulator:** A Pixel 3a API 32 arm64-v8a device is running the app. The screen displays a portrait of a man in a military uniform with a blue rectangular overlay indicating the detected face area. Below the image are buttons for "FIND TEXT" and "FIND FACE CONTOUR".
- Logcat:** The log shows TensorFlow Lite runtime initialization and model loading logs, including:

```
W/e.codelab.mlki: Unsupported class loader
I/native: face_detector_v2_jni.cc:33 Loading models/fssd_25_8bit_v1.tflite
I/native: face_detector_v2_jni.cc:33 Loading models/fssd_25_8bit_gray_v1.tflite
I/native: face_detector_v2_jni.cc:33 Loading models/LMprec_600.emd
I/native: face_detector_v2_jni.cc:33 Loading models/BCLlefteyeclosed_200.emd
I/native: face_detector_v2_jni.cc:33 Loading models/BCLRighteyeclosed_200.emd
I/native: face_detector_v2_jni.cc:33 Loading models/BCLjoy_200.emd
I/native: face_detector_v2_jni.cc:33 Loading models/MFT_fssd_fastgray.pb
I/native: face_detector_v2_jni.cc:33 Loading models/contours.tfl
I/native: face_detector_v2_jni.cc:33 Loading models/blazeface.tfl
I/tflite: Initialized TensorFlow Lite runtime.
```
- Bottom Bar:** The "Run" tab is selected, along with "TODO", "Problems", "Terminal", "Build", "Logcat", "Profiler", and "App Inspection". The status bar indicates "Launch succeeded (moments ago)".

Ondevice ML skills

For this, the app was running, I wasn't able to show any text on it to translate since emulated didn't have access to my webcam.

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project tree on the left shows files like build.gradle, TextAnalyzer.kt, MainActivity.java, and various resource folders.
- Code Editor:** The main editor window displays the `TextAnalyzer.kt` file, which contains Kotlin code for processing images and translating text.
- Device Manager:** The right-hand sidebar shows a "Virtual" tab with a "Pixel 4a API 32" device listed. The device summary indicates it's an Emulator with API 32 and 513 MB of memory.
- Emulator:** A preview of the Pixel 4a device shows a camera viewfinder with a small text overlay "Center text in box". Below the camera view, there is a text input field and a dropdown menu set to "English". A status bar at the bottom of the screen says "Waiting for text recognition in second selected language".
- Build Output:** The bottom pane shows the build log with the message "Launch succeeded". The log details the build process, including tasks like `:app:mergeDebugJniLibFolders`, `:app:stripDebugDebugSymbols`, and `:app:packageDebug`, and concludes with "BUILD SUCCESSFUL in 2s".

Ondevice ML skills