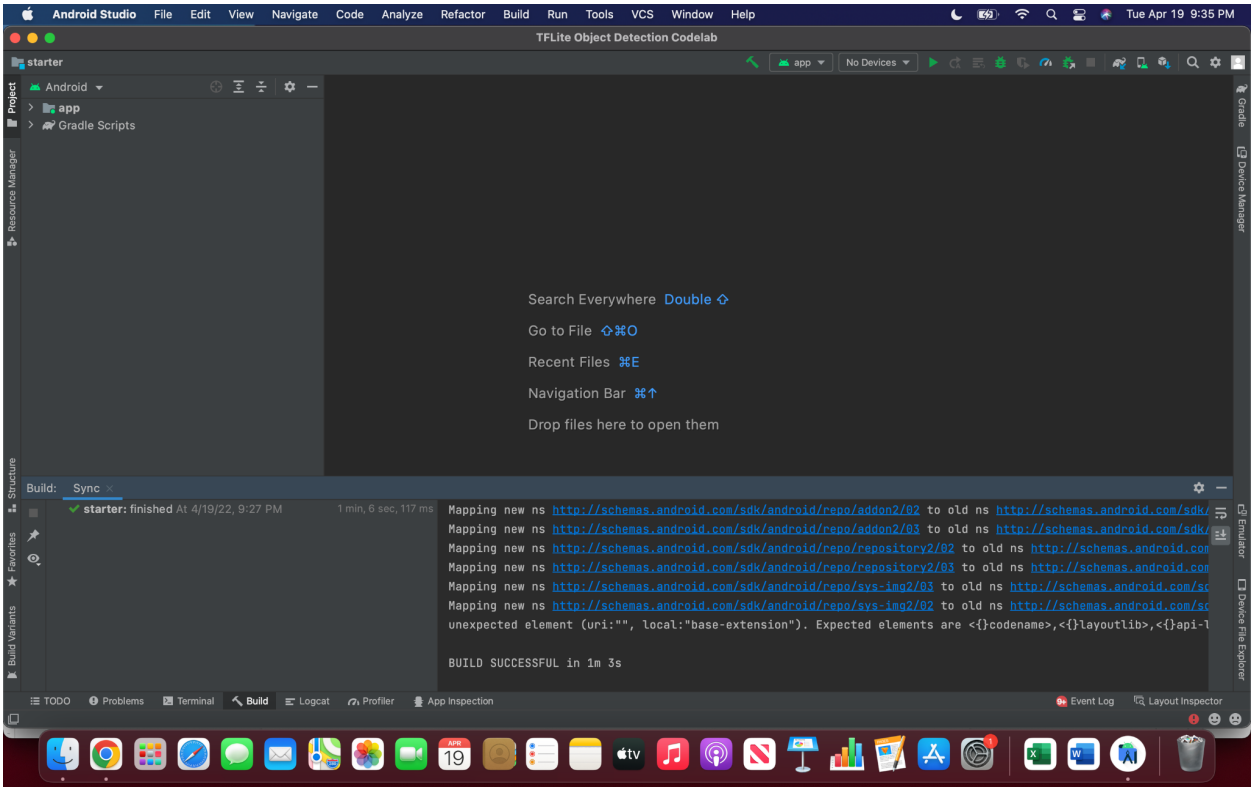
CMPE 258 – ASSIGNMENT 5 PART 2
Advanced Keras and Intro to edge ML

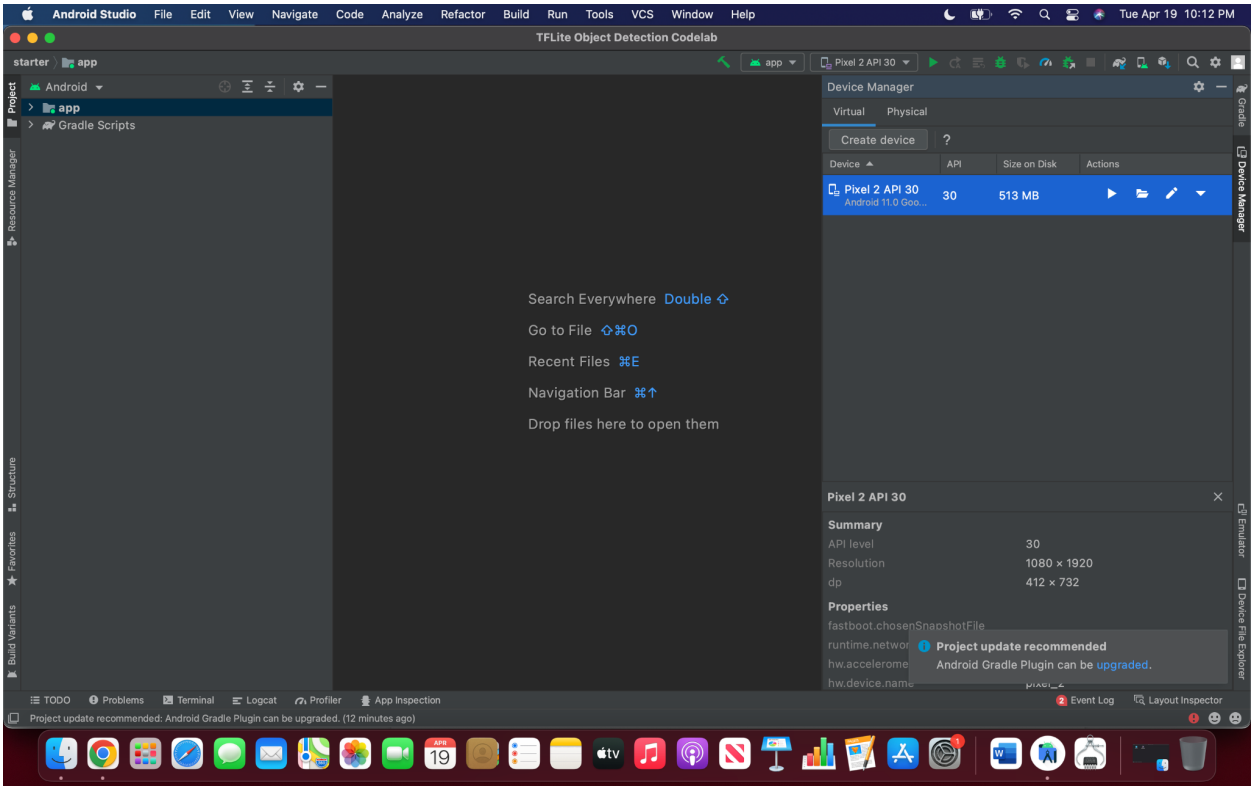
SUBMITTED BY: SWATHI ANANDRAM

A) Mobile App and Image training on device: Build and deploy a custom object detection model using tflite on android simulator

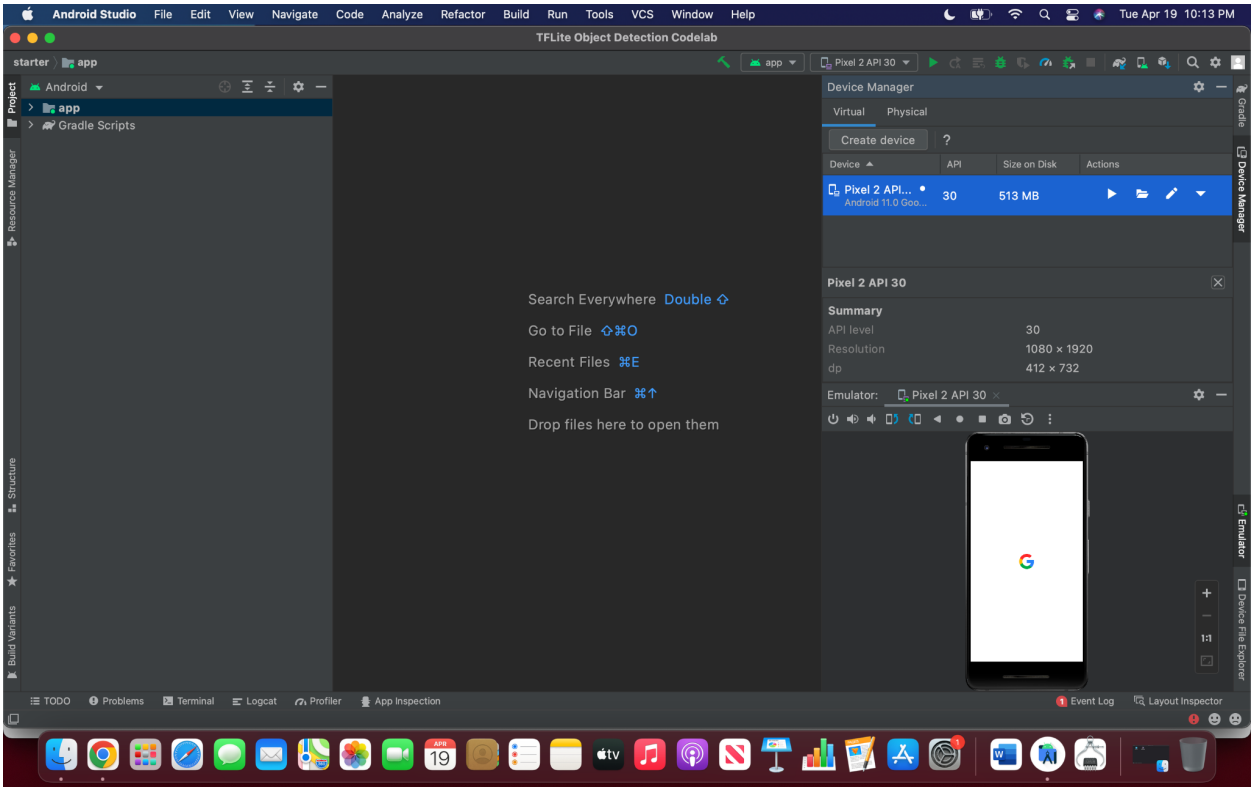
- Download Android Studio and Import Started Project



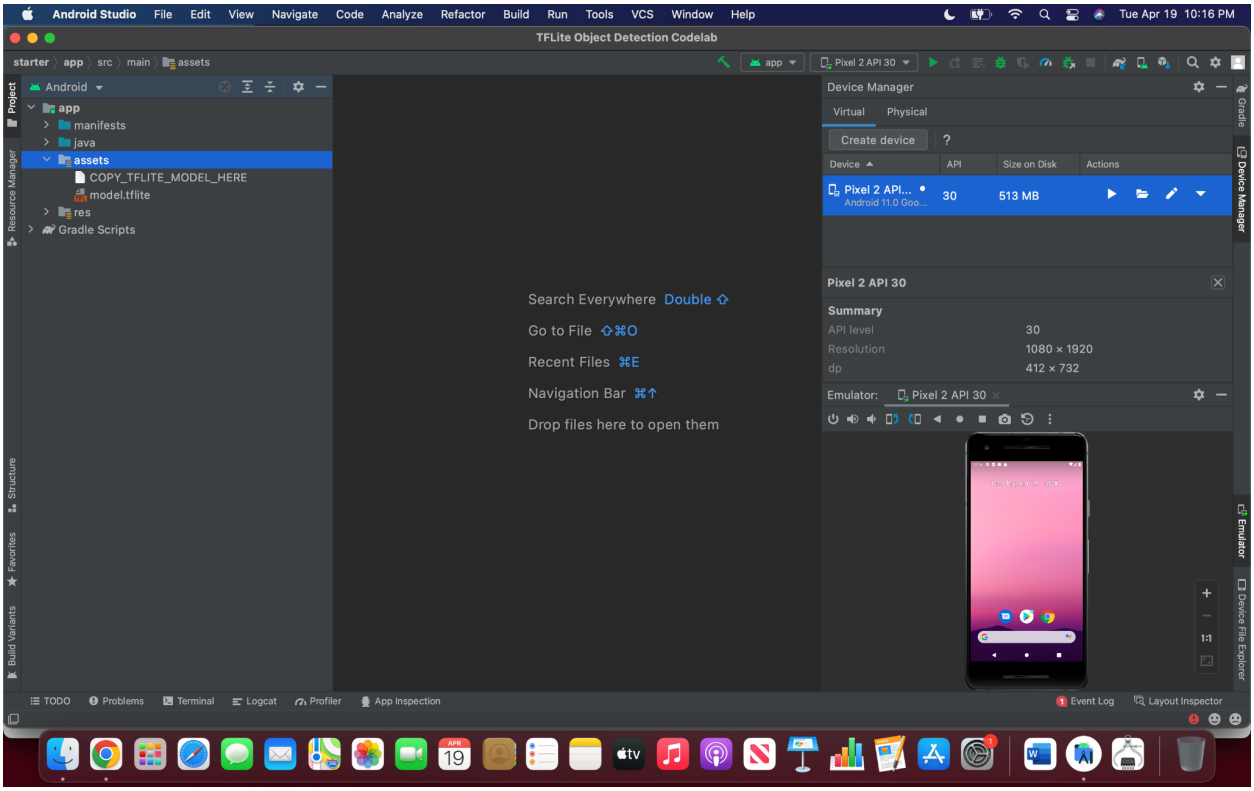
- Adding Virtual Device



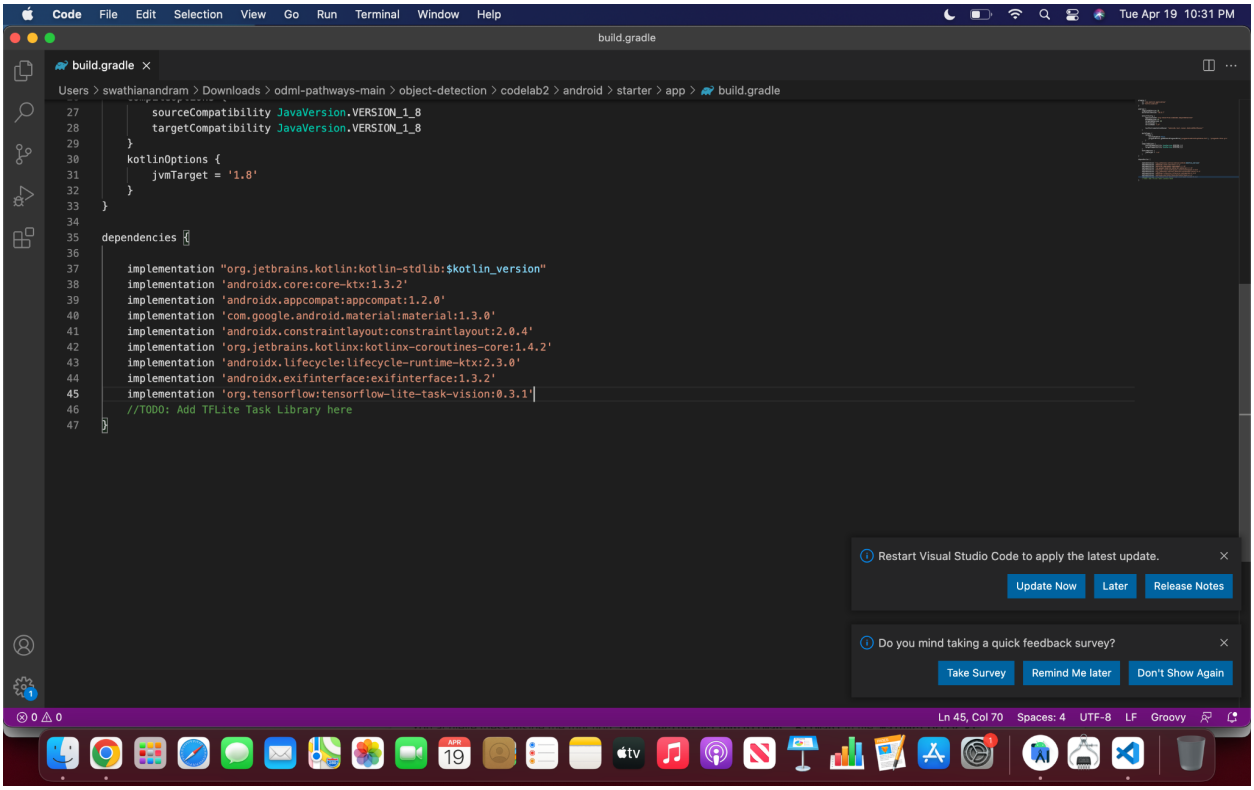
- Run Virtual Device



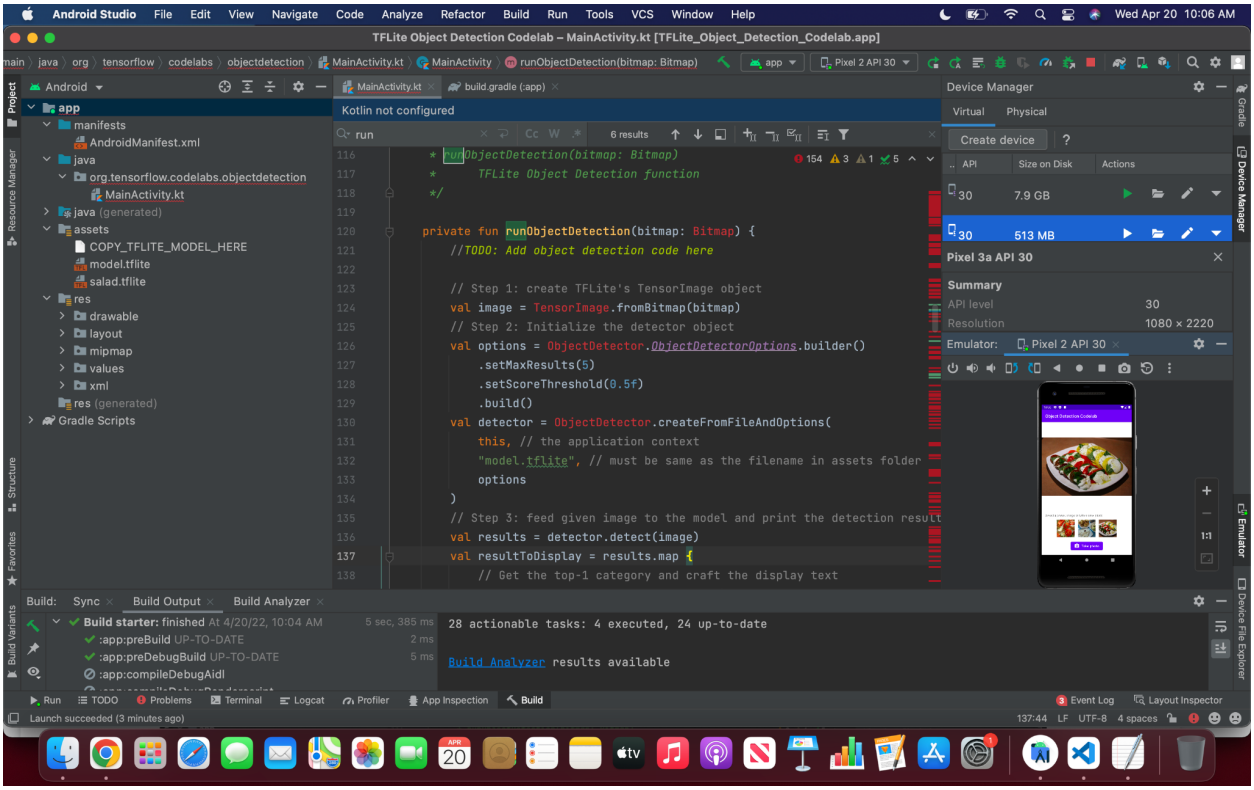
- Adding the model to the starter project

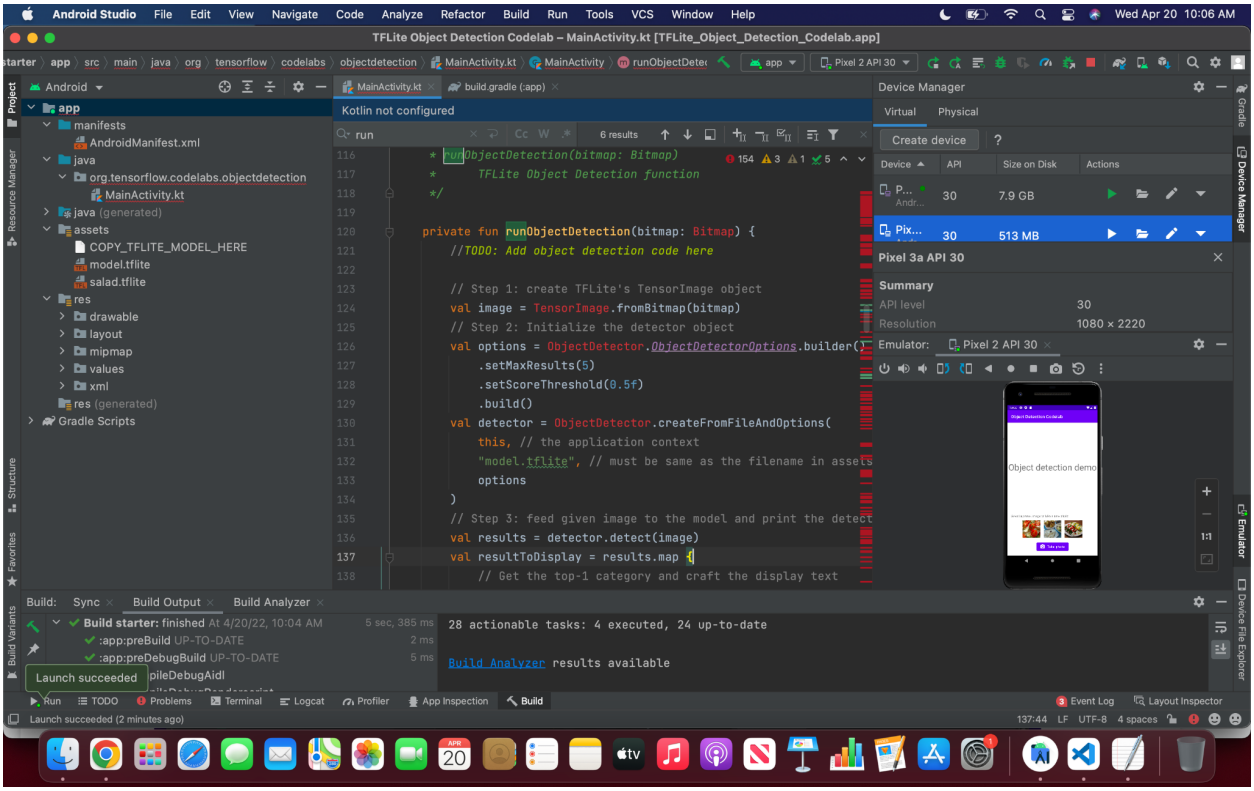


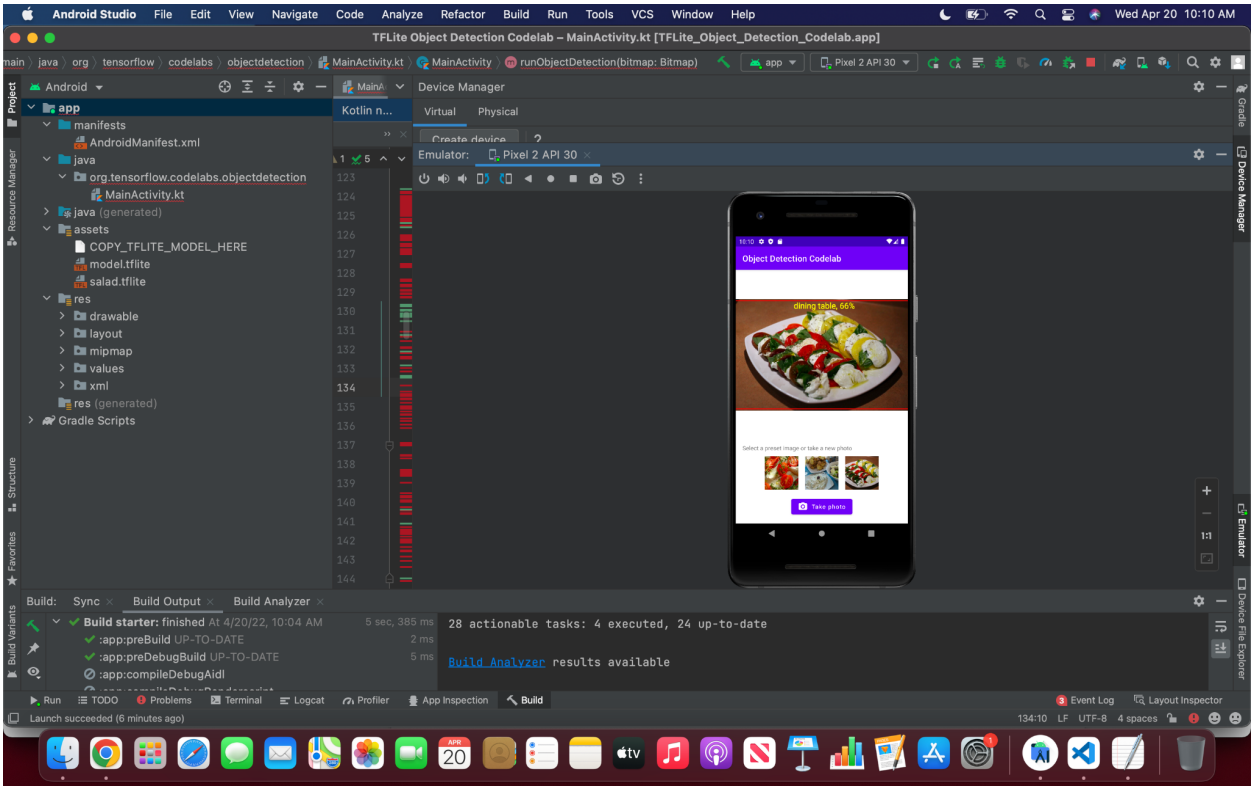
- Adding dependency in build.gradle

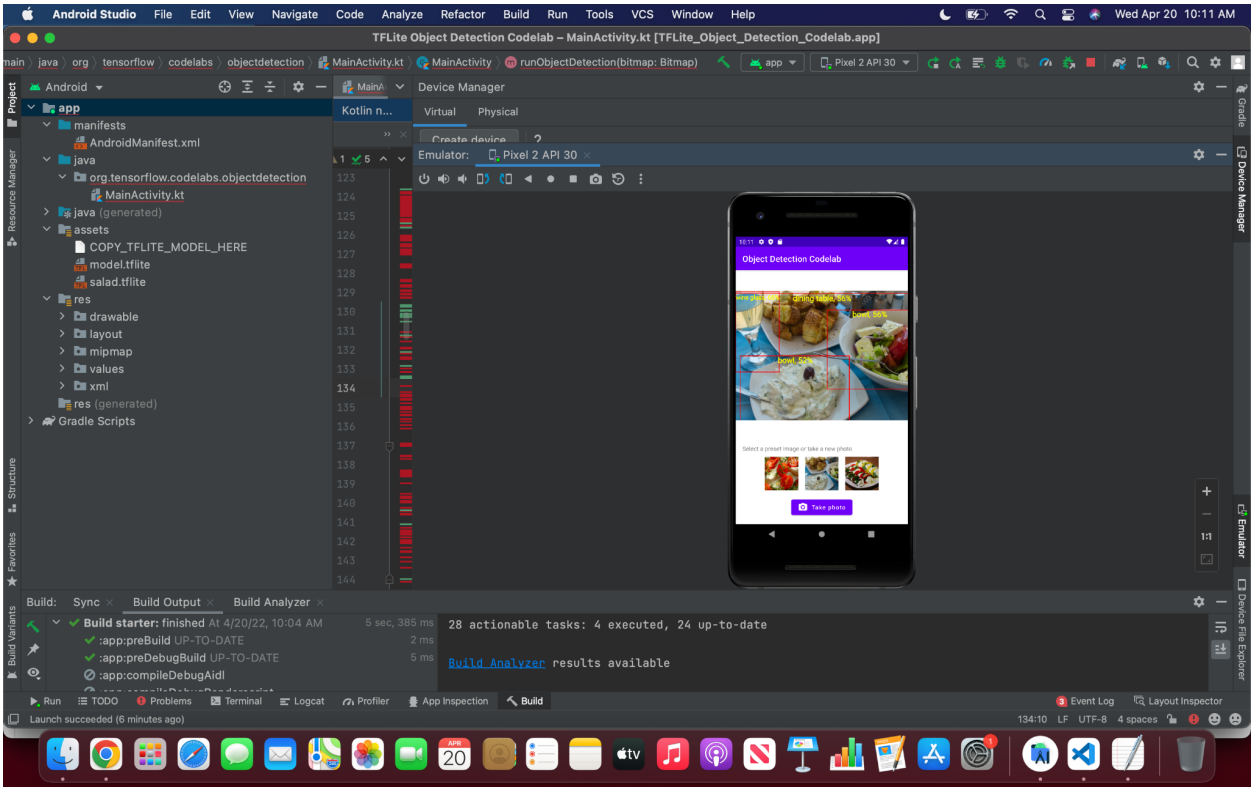


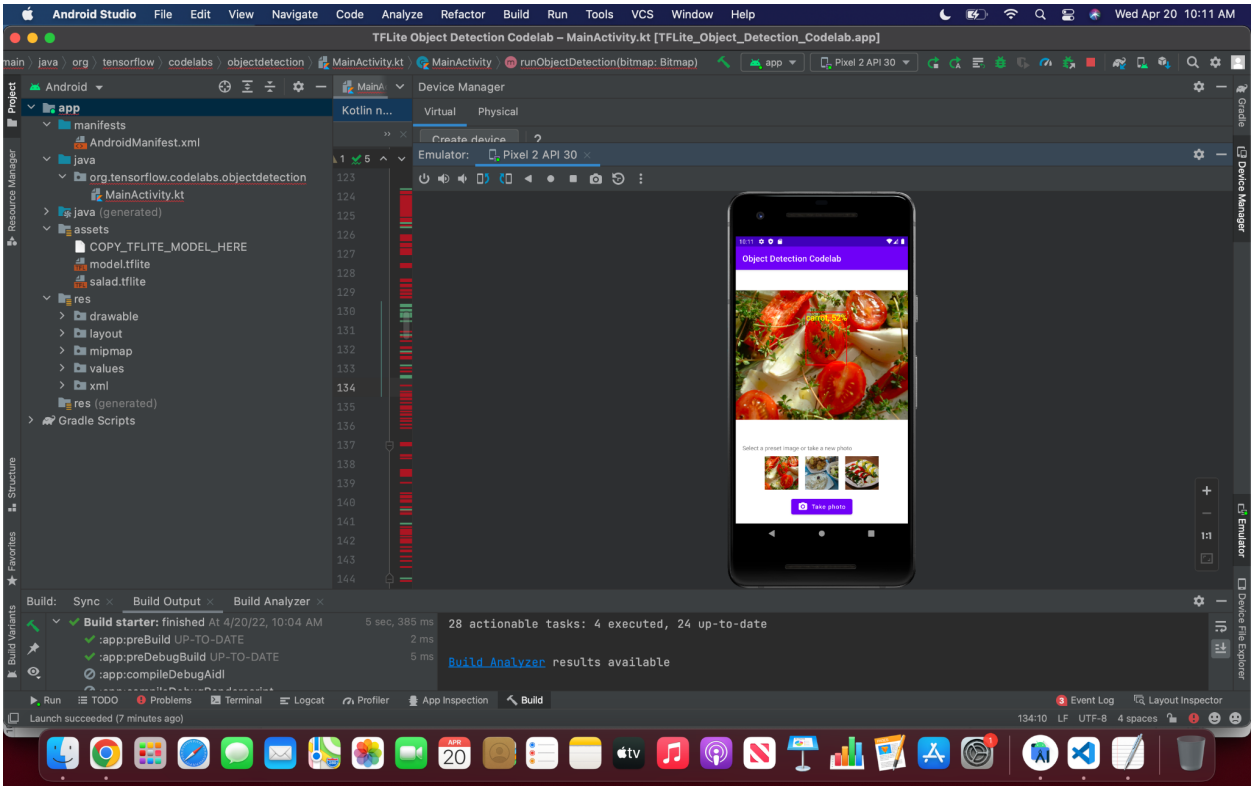
- Create Image Object, Create a Detector instance , Feed Image(s) to the detector


Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Wed Apr 20 10:06 AM
TFLite Object Detection Codelab – MainActivity.kt [TFLite_Object_Detection_Codelab.app]
Kotlin not configured
Device Manager
Virtual
Physical
Create device
Pixel 3a API 30
Summary
API level 30
Resolution 1080 × 2220
Emulator: Pixel 2 API 30
Build: Sync Build Output Build Analyzer
Build starter: finished At 4/20/22, 10:04 AM 5 sec, 385 ms
:app:preBuild UP-TO-DATE 2 ms
:app:preDebugBuild UP-TO-DATE 5 ms
:app:compileDebugAidl
28 actionable tasks: 4 executed, 24 up-to-date
Build Analyzer results available
Launch succeeded (3 minutes ago)
137:44 LF UTF-8 4 spaces

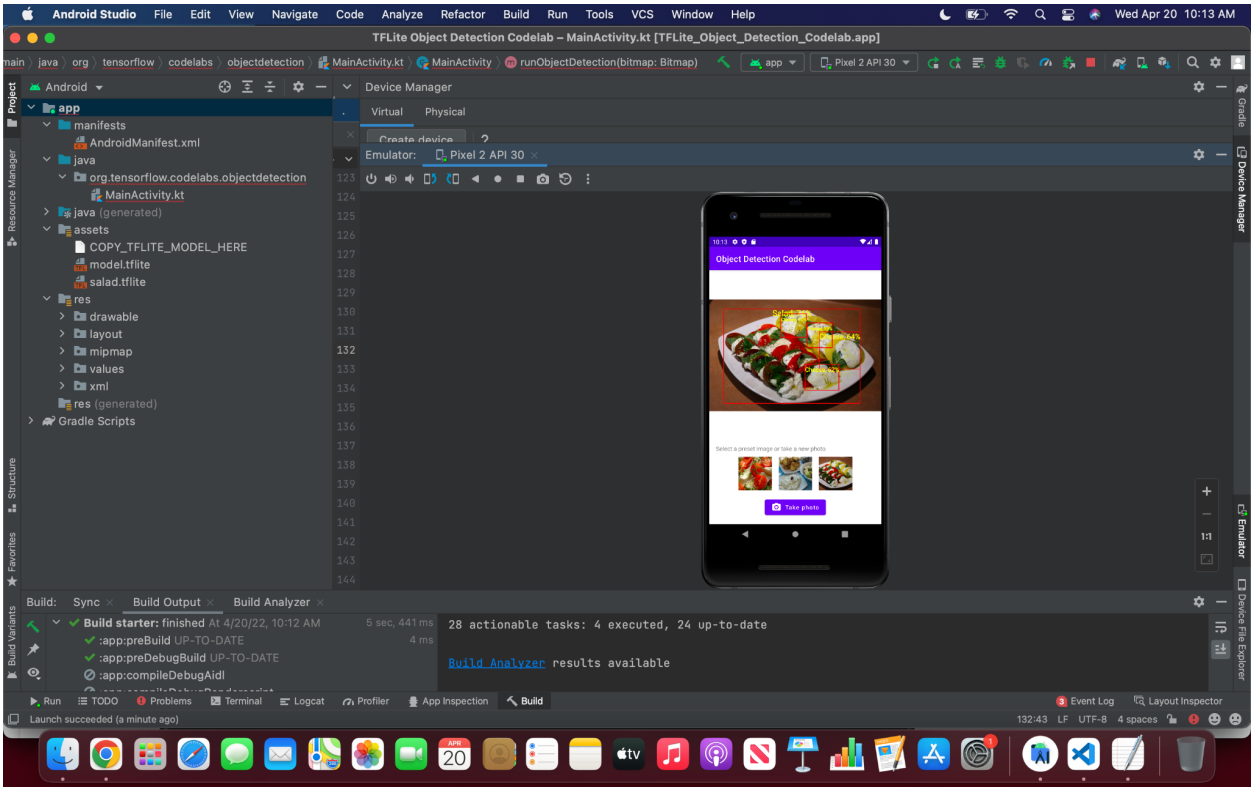

Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Wed Apr 20 10:06 AM
TFLite Object Detection Codelab – MainActivity.kt [TFLite_Object_Detection_Codelab.app]
Kotlin not configured
Device Manager
Virtual
Physical
Create device
Pixel 3a API 30
Summary
API level 30
Resolution 1080 × 2220
Emulator: Pixel 2 API 30
Object detection demo
Build: Sync Build Output Build Analyzer
Build starter: finished At 4/20/22, 10:04 AM 5 sec, 385 ms
:app:preBuild UP-TO-DATE 2 ms
:app:preDebugBuild UP-TO-DATE 5 ms
Launch succeeded
28 actionable tasks: 4 executed, 24 up-to-date
Build Analyzer results available
Launch succeeded (2 minutes ago)
137:44 LF UTF-8 4 spaces


Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Wed Apr 20 10:10 AM
TFLite Object Detection Codelab – MainActivity.kt [TFLite_Object_Detection_Codelab.app]
Device Manager
Virtual
Physical
Emulator: Pixel 2 API 30
Object Detection Codelab
dining table, 66%
Take photo
Build: Sync Build Output Build Analyzer
Build starter: finished At 4/20/22, 10:04 AM 5 sec, 385 ms
:app:preBuild UP-TO-DATE 2 ms
:app:preDebugBuild UP-TO-DATE 5 ms
:app:compileDebugAidl
28 actionable tasks: 4 executed, 24 up-to-date
Build Analyzer results available
Launch succeeded (6 minutes ago)
134:10 LF UTF-8 4 spaces

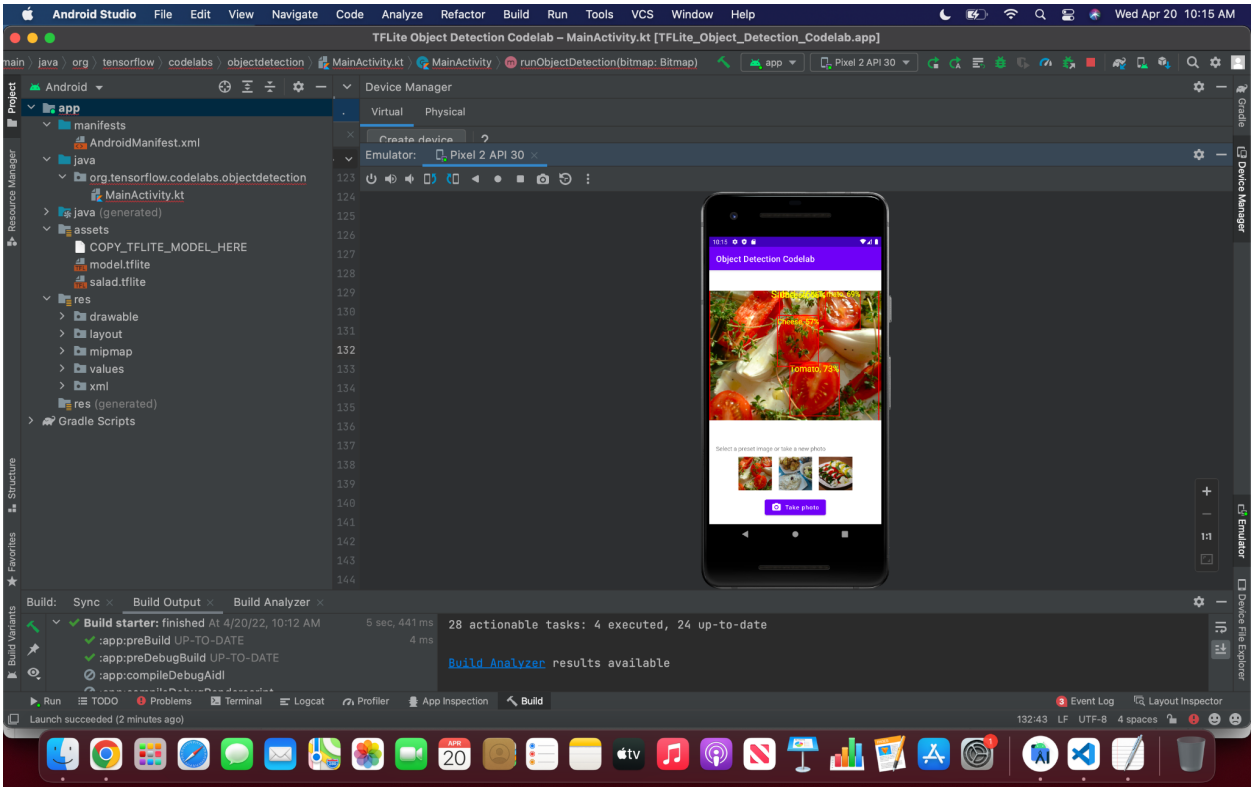

Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Wed Apr 20 10:11 AM
TFLite Object Detection Codelab – MainActivity.kt [TFLite_Object_Detection_Codelab.app]
Device Manager
Virtual
Physical
Emulator: Pixel 2 API 30
Object Detection Codelab
dining table, 66%
bowl, 66%
bowl, 52%
Take photo
Build: Sync Build Output Build Analyzer
Build starter: finished At 4/20/22, 10:04 AM 5 sec, 385 ms
:app:preBuild UP-TO-DATE 2 ms
:app:preDebugBuild UP-TO-DATE 5 ms
:app:compileDebugAidl
28 actionable tasks: 4 executed, 24 up-to-date
Build Analyzer results available
Launch succeeded (6 minutes ago)
134:10 LF UTF-8 4 spaces

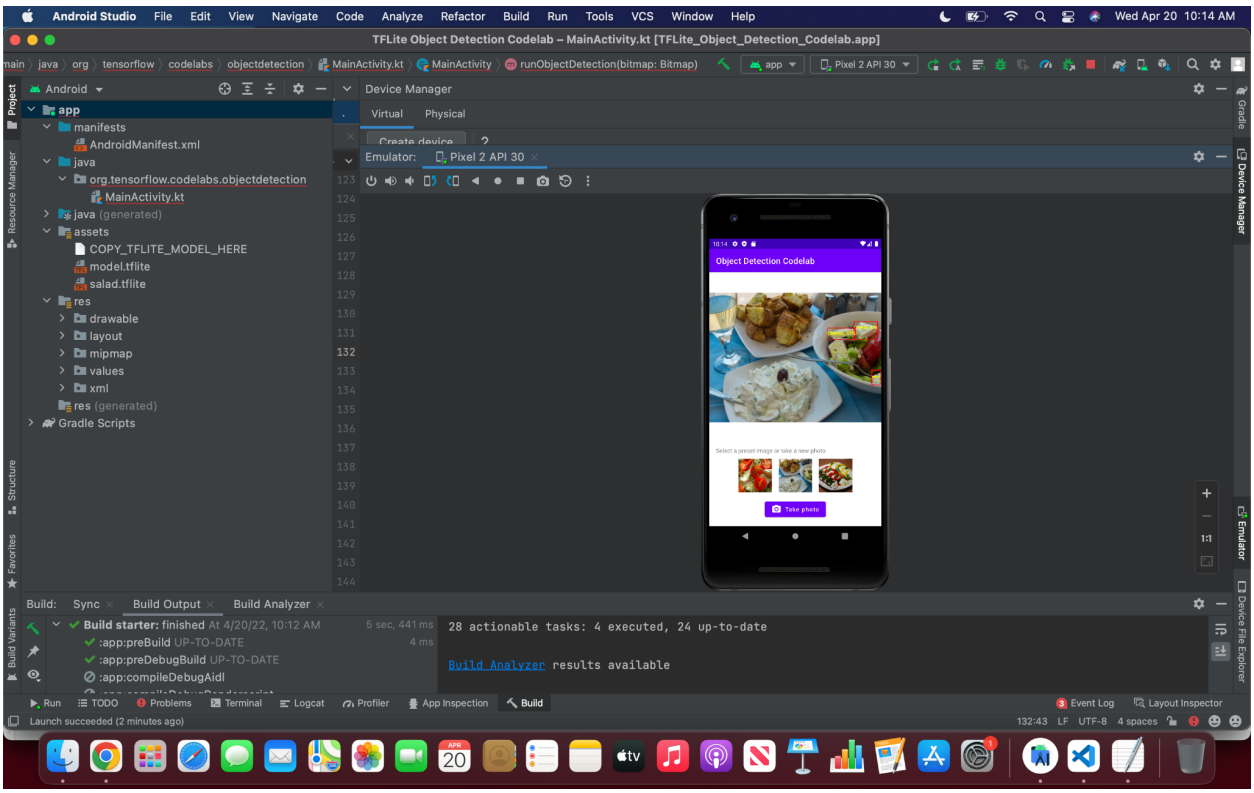


- Changing model



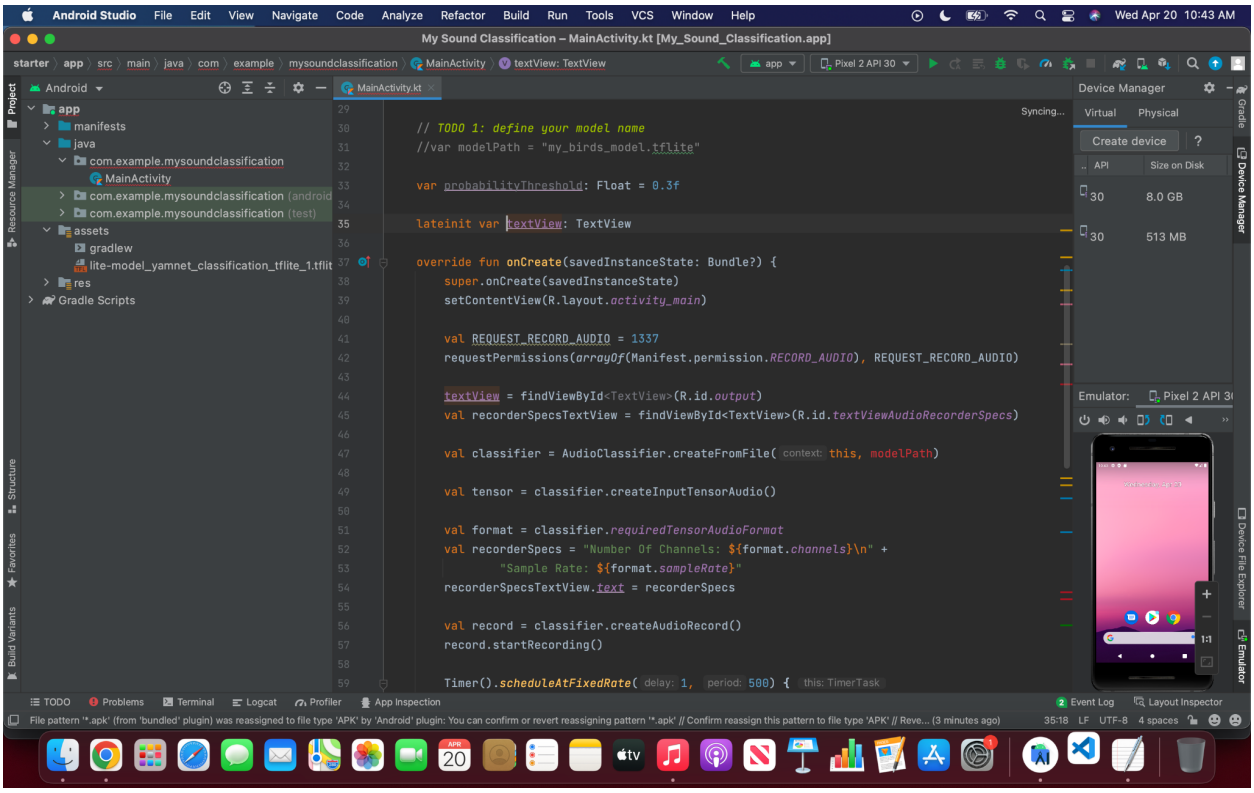
As we can see the bounding boxes are able to recognize various elements on the plate unlike the previous model which detected it as dining table

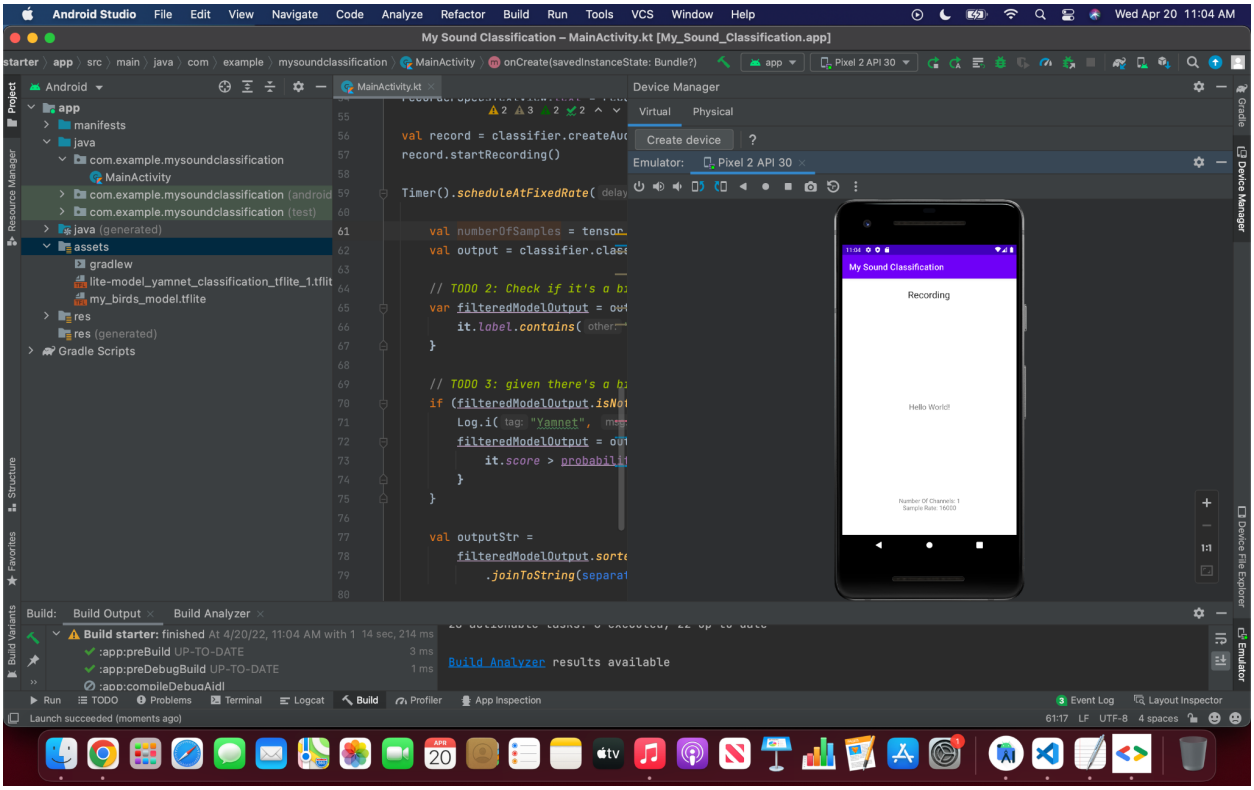




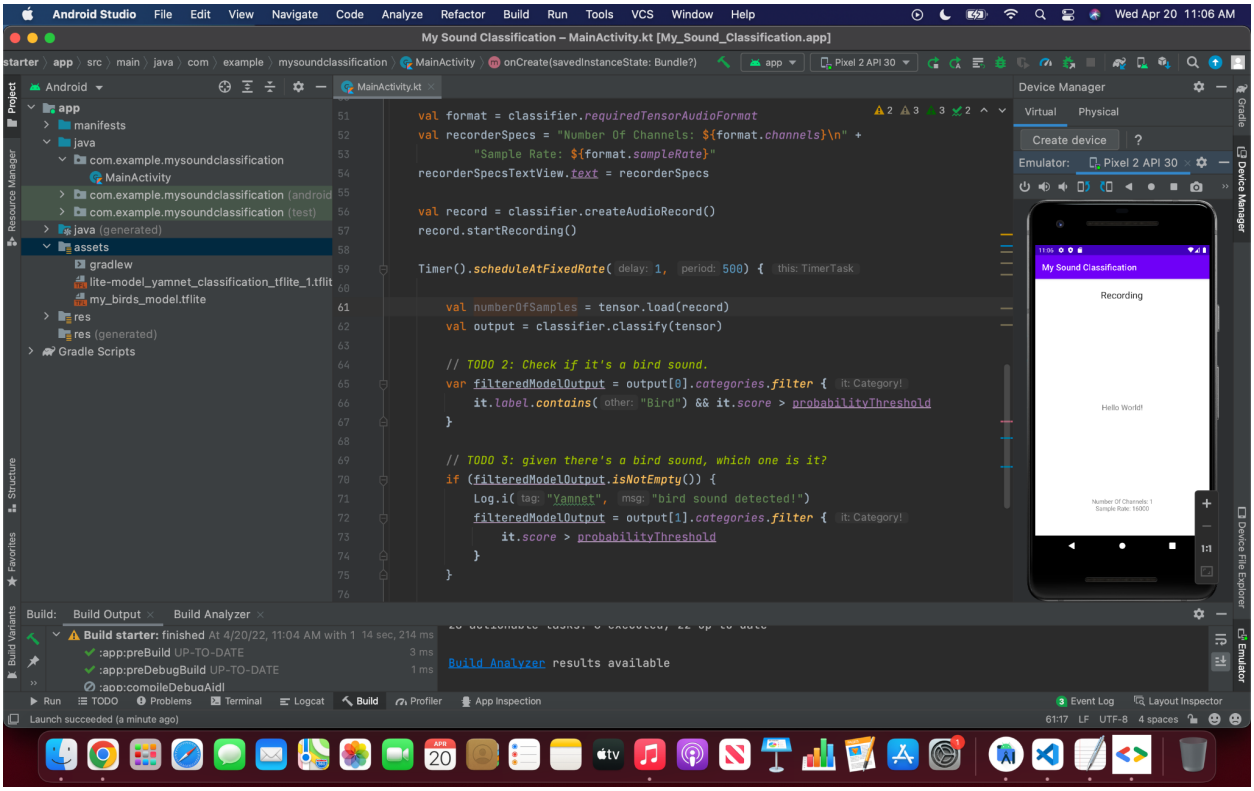
B) Mobile app and audio training on device : Build a custom pre-trained Audio Classification model

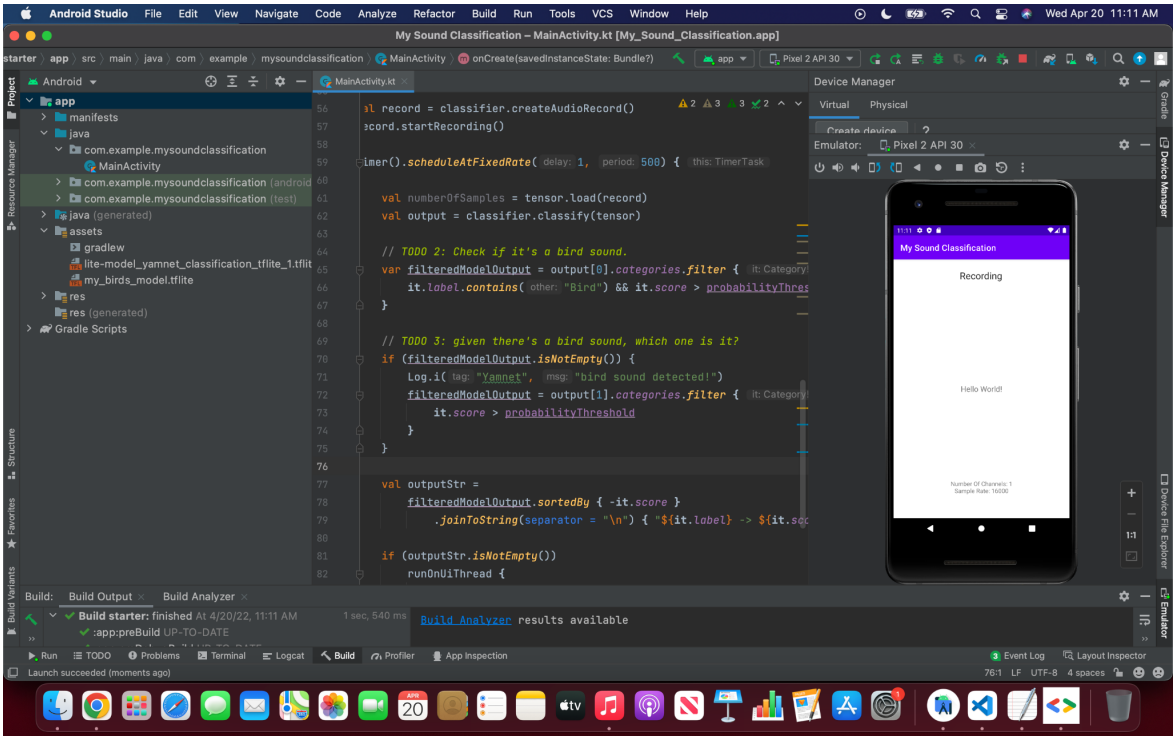
- Importing the starter project and starting the emulator, Loading the model, Creating the audio recorder and beginning the recording



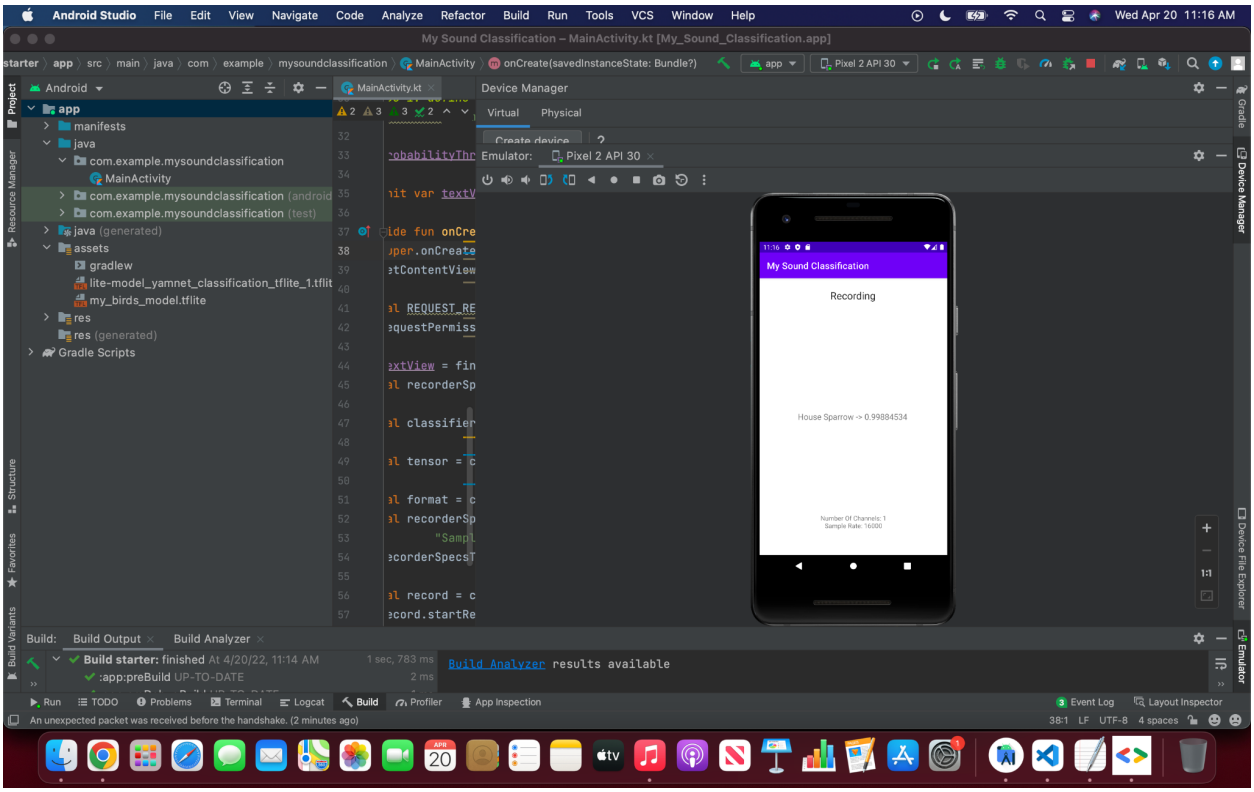


- Change the filtering to also filter out anything that is not Bird



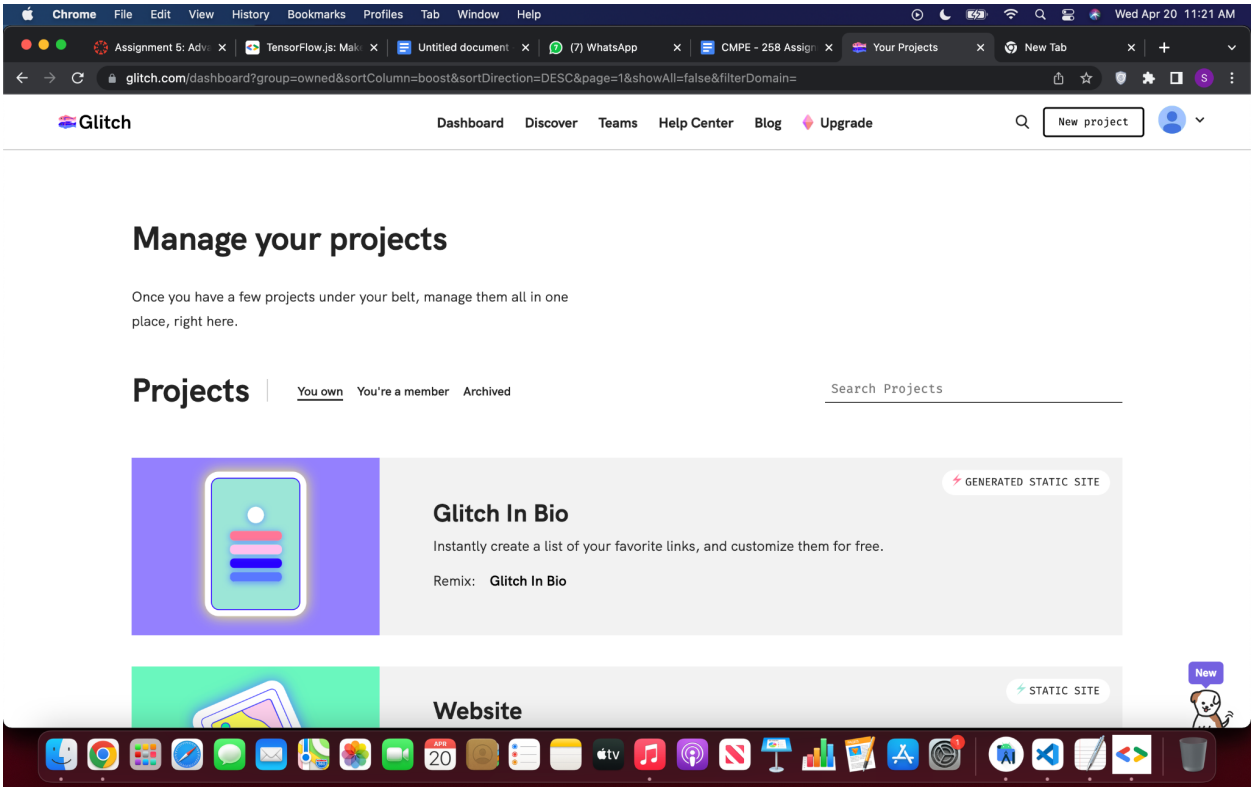


- Predicted Results when played a sparrow sound which means the model is working fine

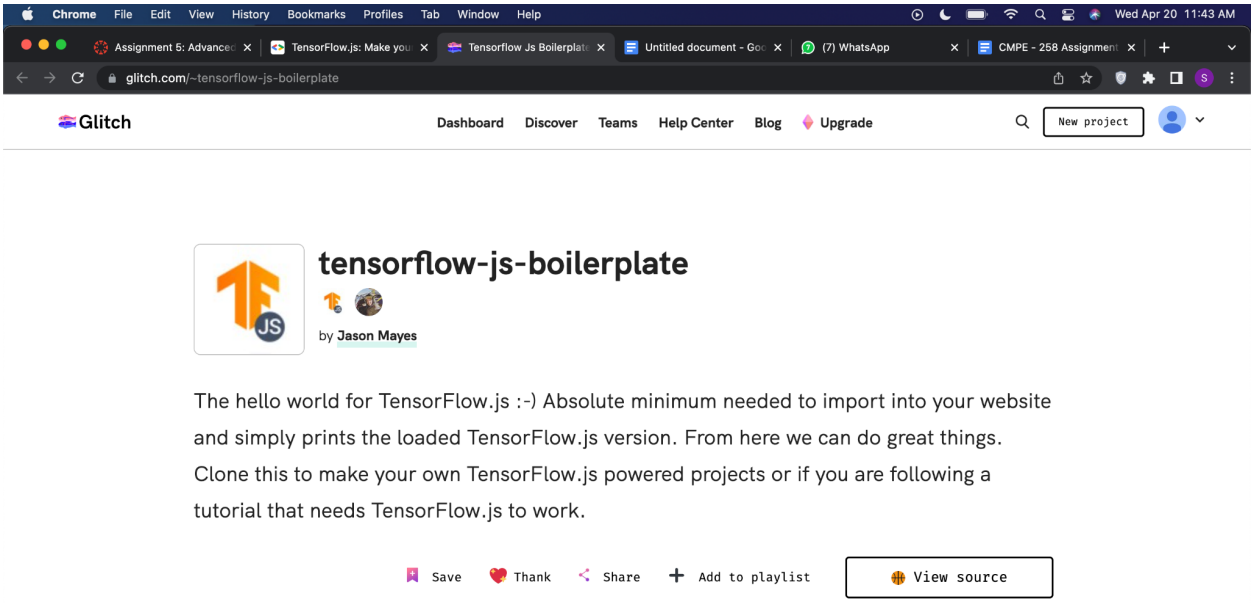


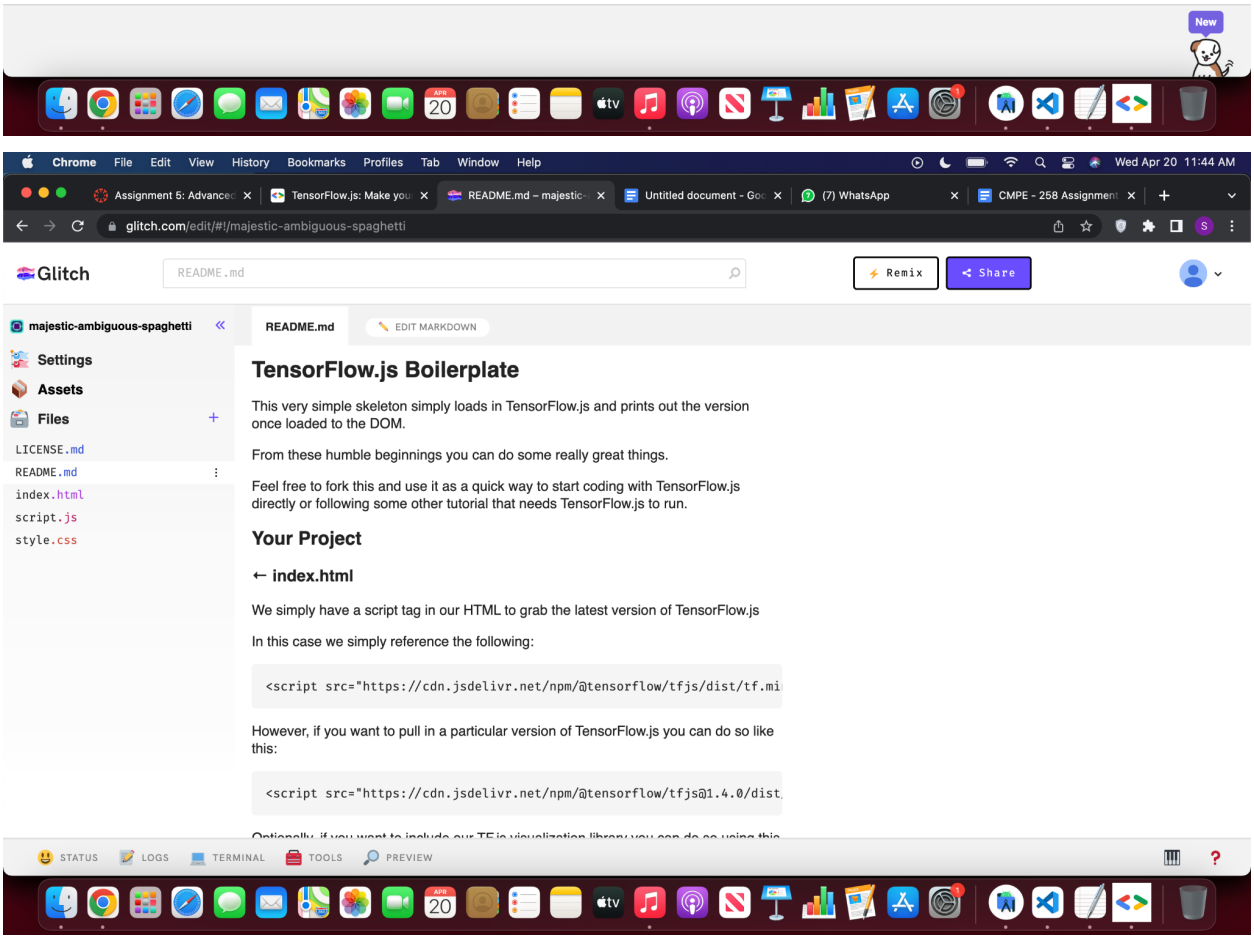
C) Web app and image training on device: TensorFlow.js Transfer Learning Image Classifier

- Create a Glitch profile

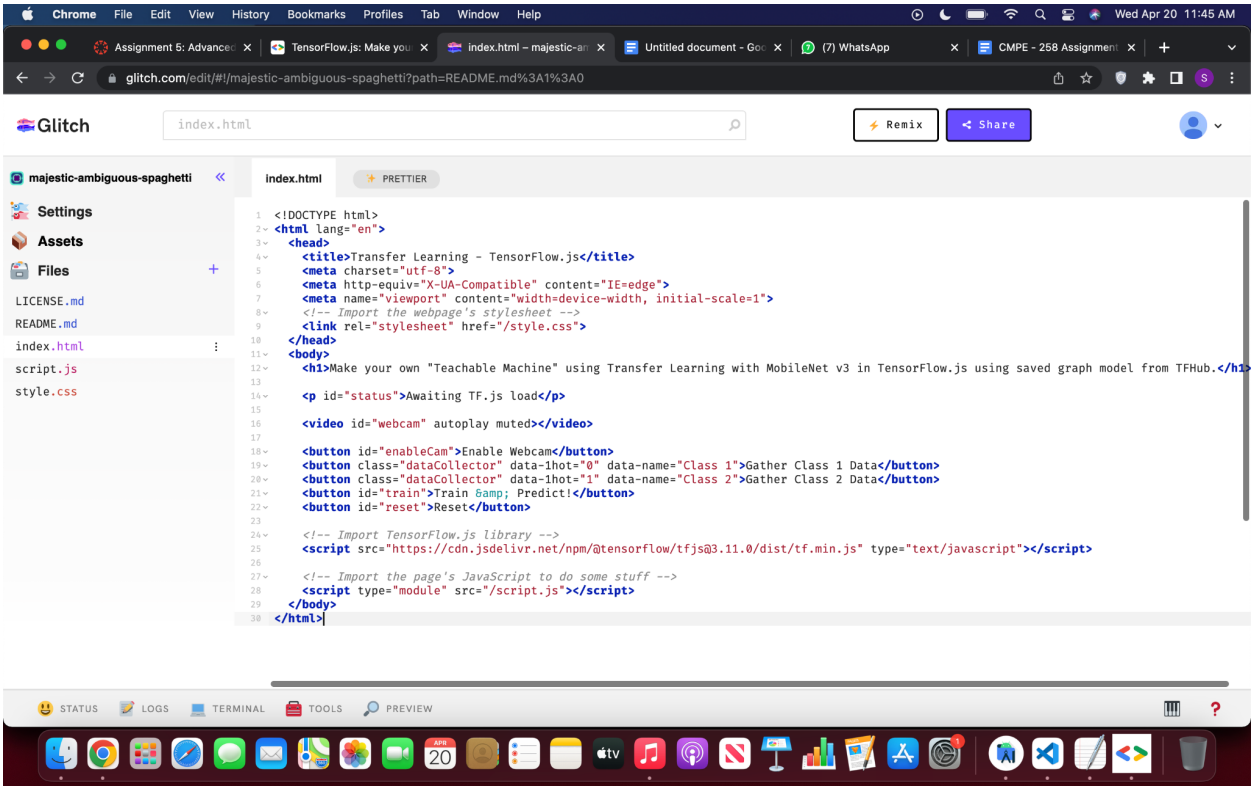


- Get the boilerplate template

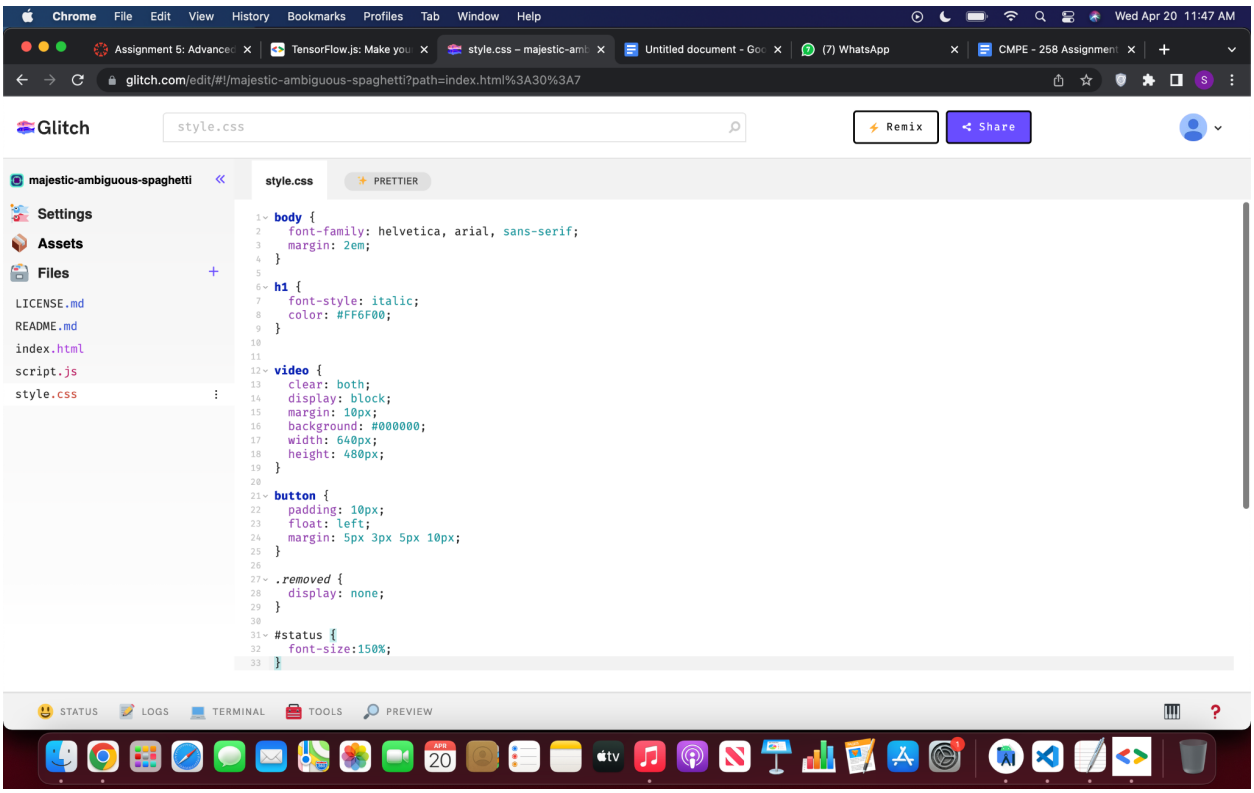




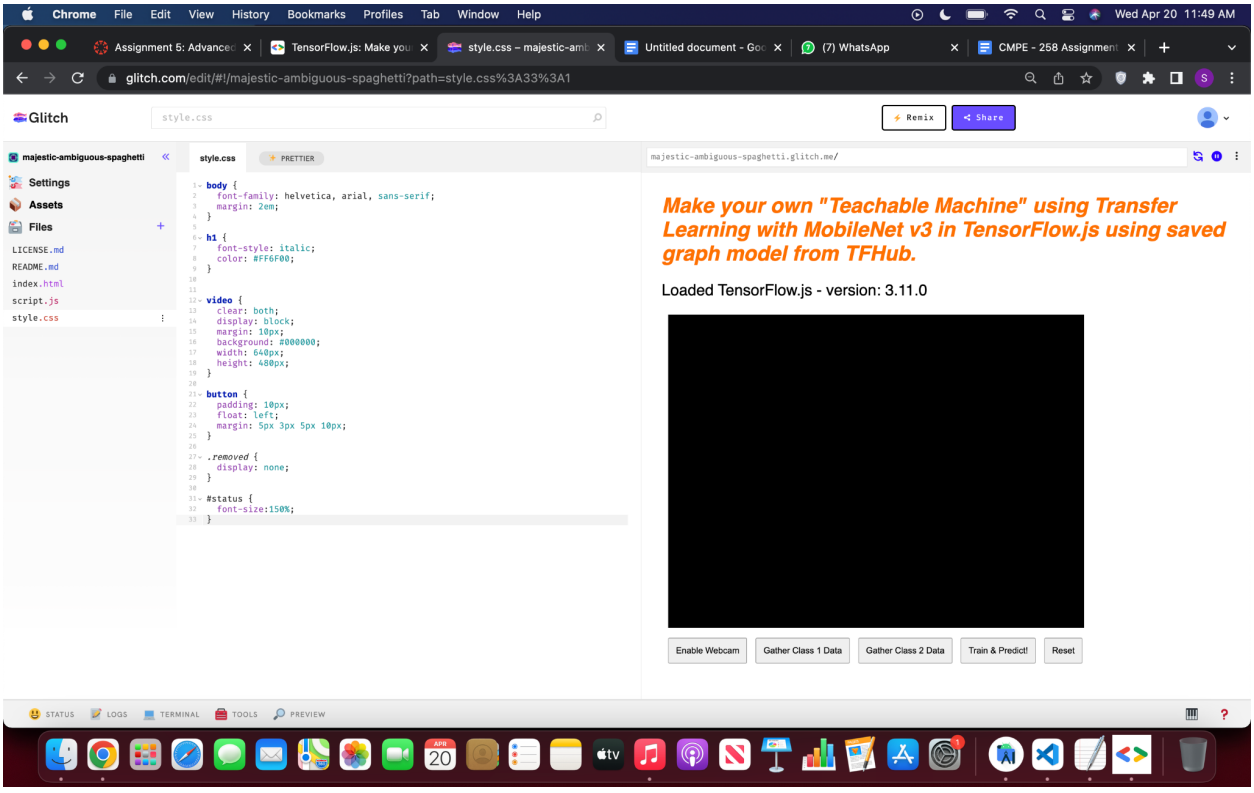
- APP HTML boilerplate changes


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Apr 20 11:45 AM
glitch.com/edit/#!/majestic-ambiguous-spaghetti?path=README.md%3A1%3A0
Glitch
index.html
Remix
Share
majestic-ambiguous-spaghetti
Settings
Assets
Files
LICENSE.md
README.md
index.html
script.js
style.css
index.html
PRETTIER
<!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">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Import the webpage's stylesheet -->
<link rel="stylesheet" href="/style.css">
</head>
<body>
<h1>Make your own "Teachable Machine" using Transfer Learning with MobileNet v3 in TensorFlow.js using saved graph model from TFHub.</h1>
Awaiting TF.js load
<video id="webcam" autoplay muted></video>
<button id="enableCam">Enable Webcam</button>
<button class="dataCollector" data-1hot="0" data-name="Class 1">Gather Class 1 Data</button>
<button class="dataCollector" data-1hot="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" type="text/javascript"></script>
<!-- Import the page's JavaScript to do some stuff -->
<script type="module" src="/script.js"></script>
</body>
</html>
STATUS LOGS TERMINAL TOOLS PREVIEW

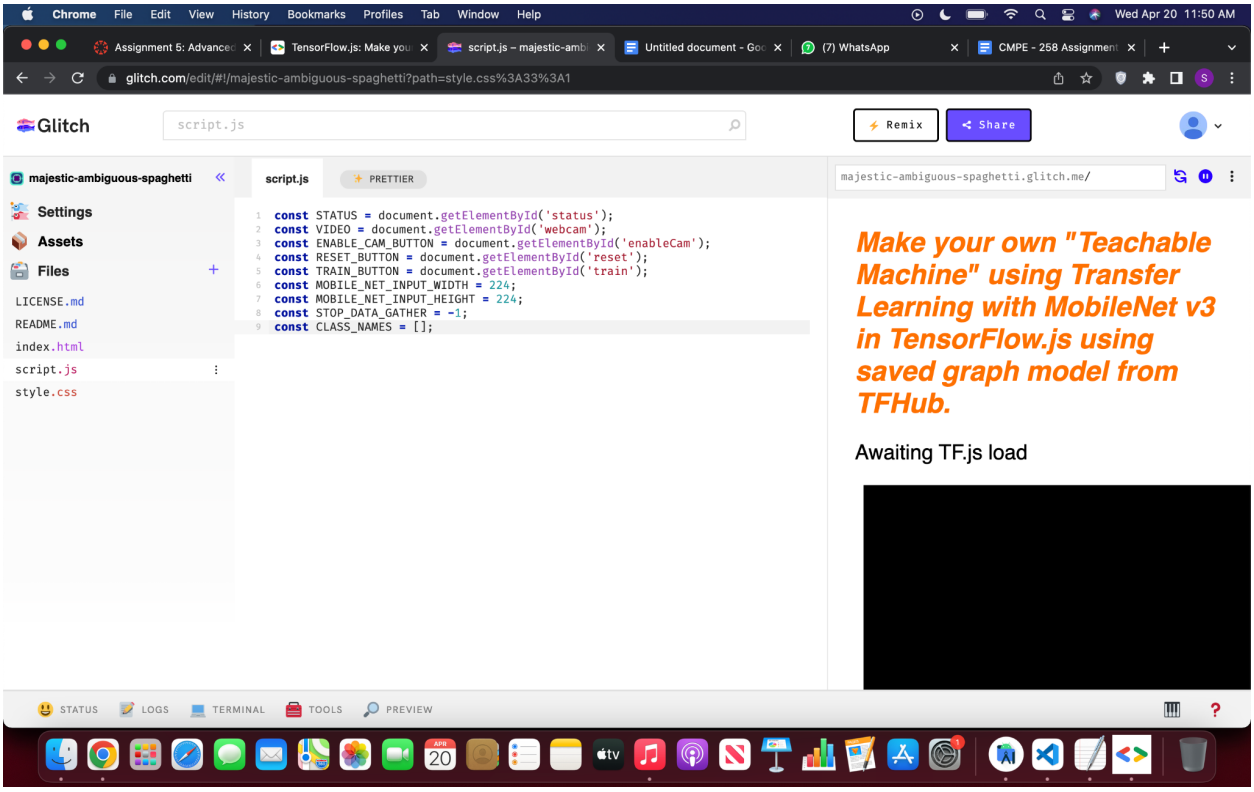
- Add styles


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Apr 20 11:47 AM
glitch.com/edit/#!/majestic-ambiguous-spaghetti?path=index.html%3A30%3A7
Glitch
style.css
Remix
Share
majestic-ambiguous-spaghetti
Settings
Assets
Files
LICENSE.md
README.md
index.html
script.js
style.css
style.css
PRETTIER
body {
font-family: helvetica, arial, sans-serif;
margin: 2em;
}
h1 {
font-style: italic;
color: #FF6F00;
}
video {
clear: both;
display: block;
margin: 10px;
background: #000000;
width: 640px;
height: 480px;
}
button {
padding: 10px;
float: left;
margin: 5px 3px 5px 10px;
}
.removed {
display: none;
}
#status {
font-size:150%;
}
STATUS LOGS TERMINAL TOOLS PREVIEW

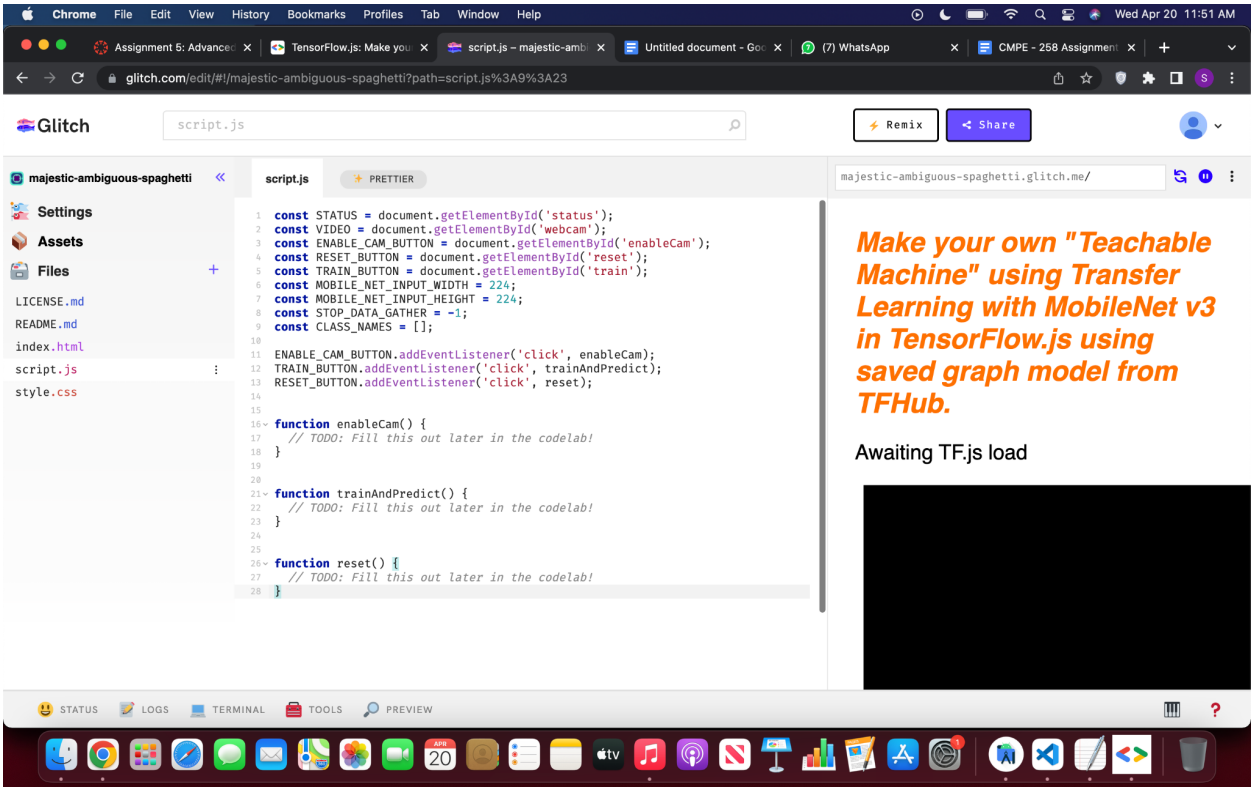
- Preview the changes



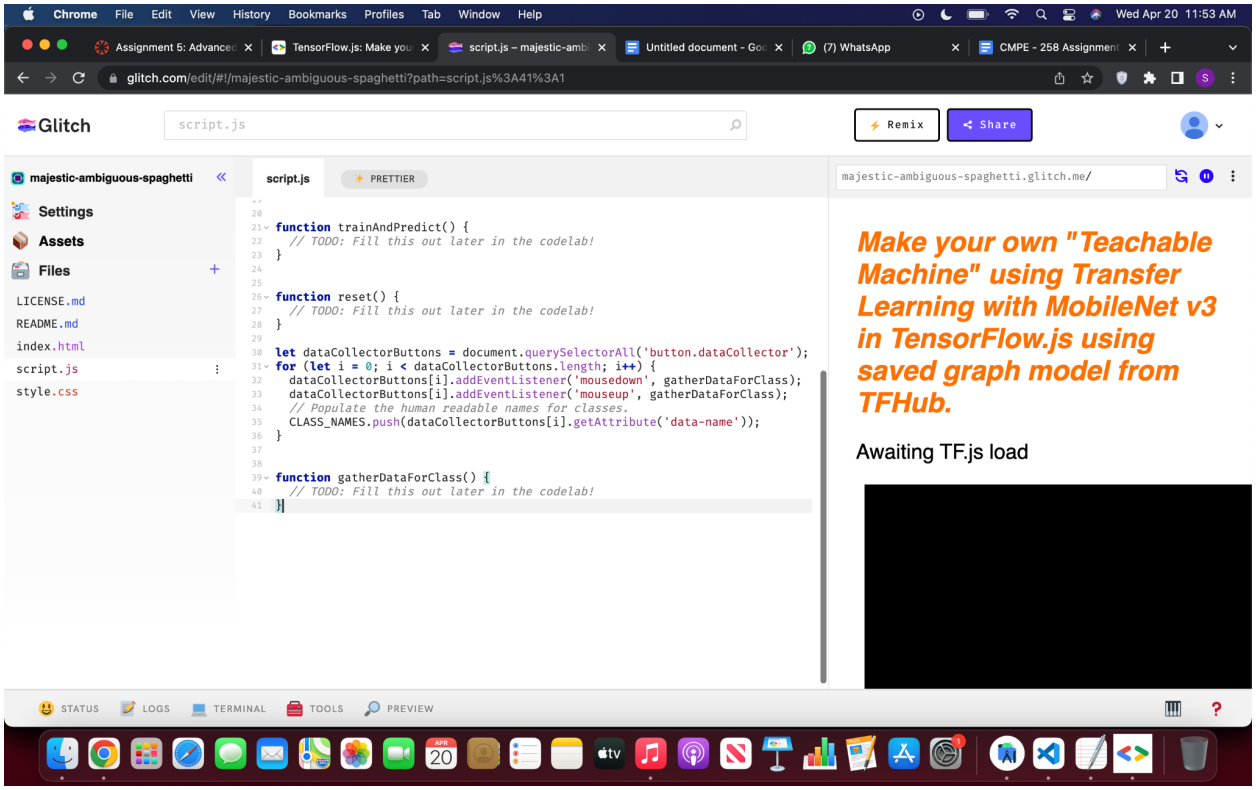
- Define Key Constants



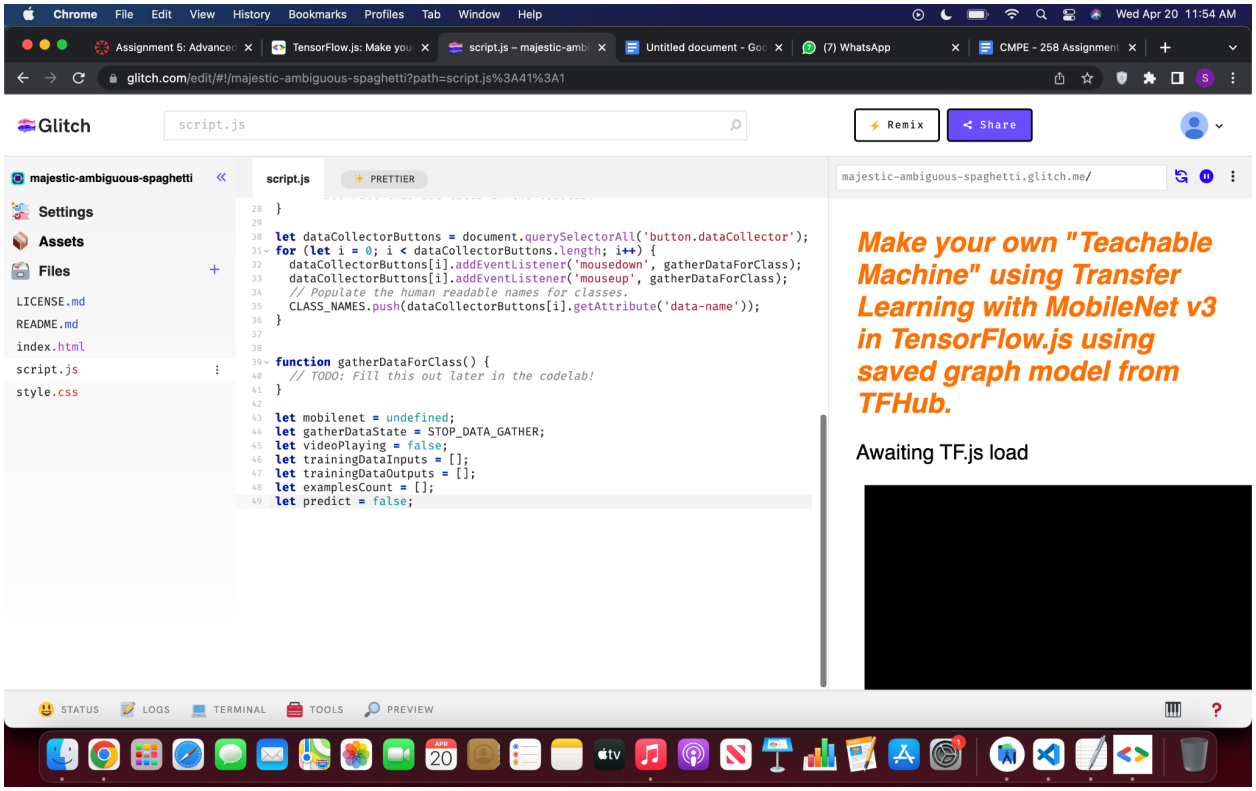
- Add key event listeners



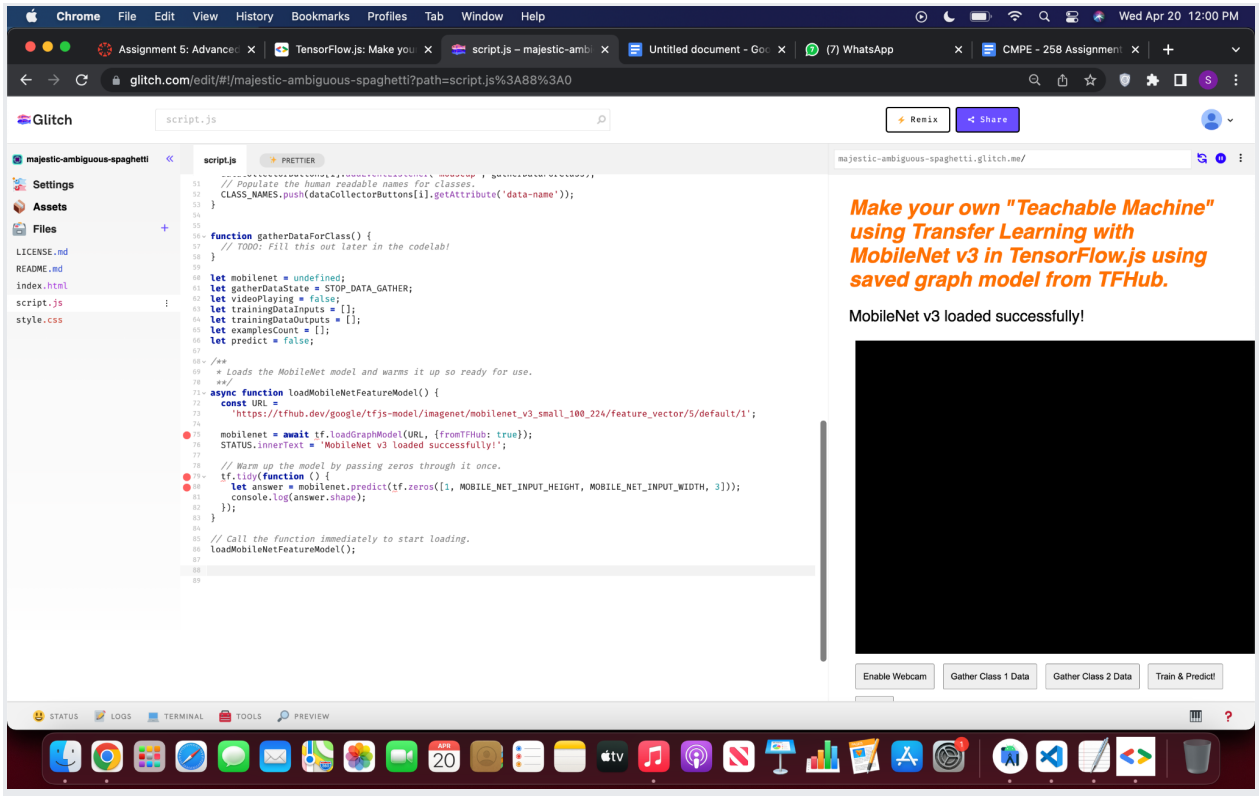
- you can find all buttons that have a class of ‘dataCollector' using document.querySelectorAll()



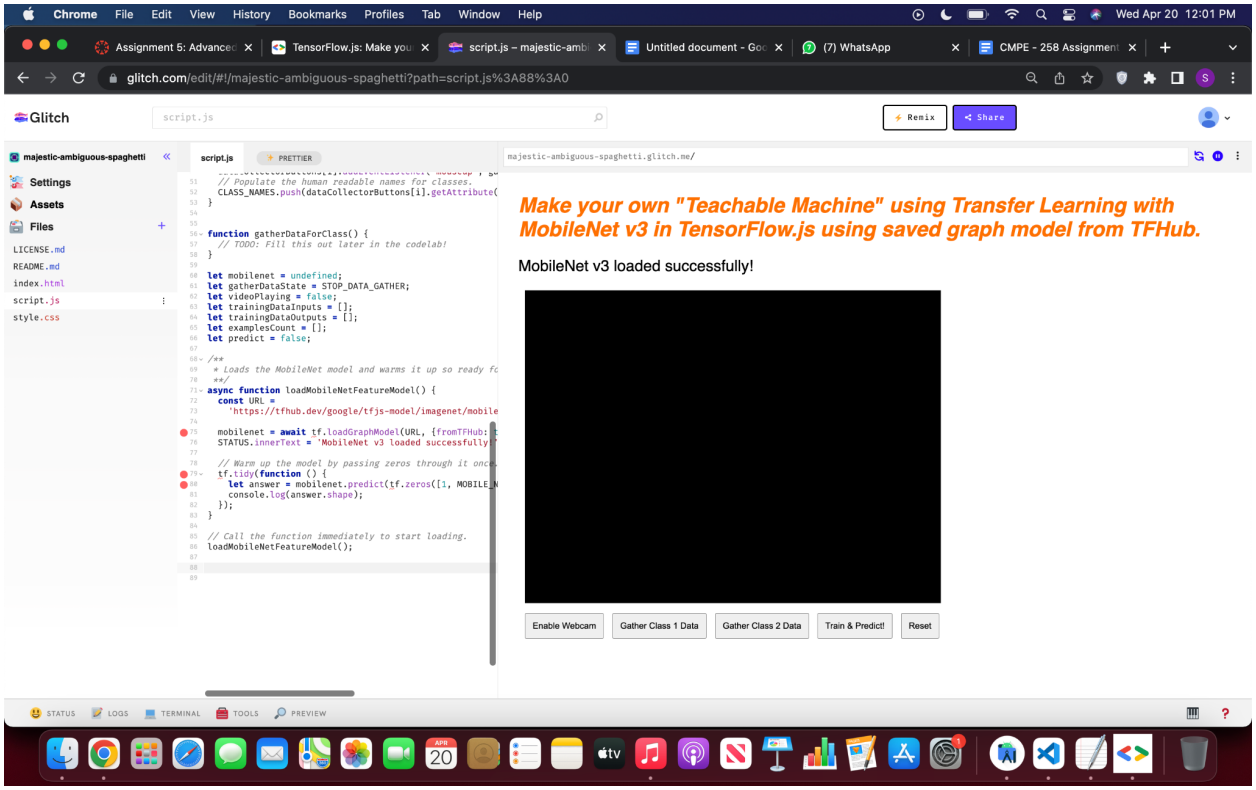
- add some variables to store key things



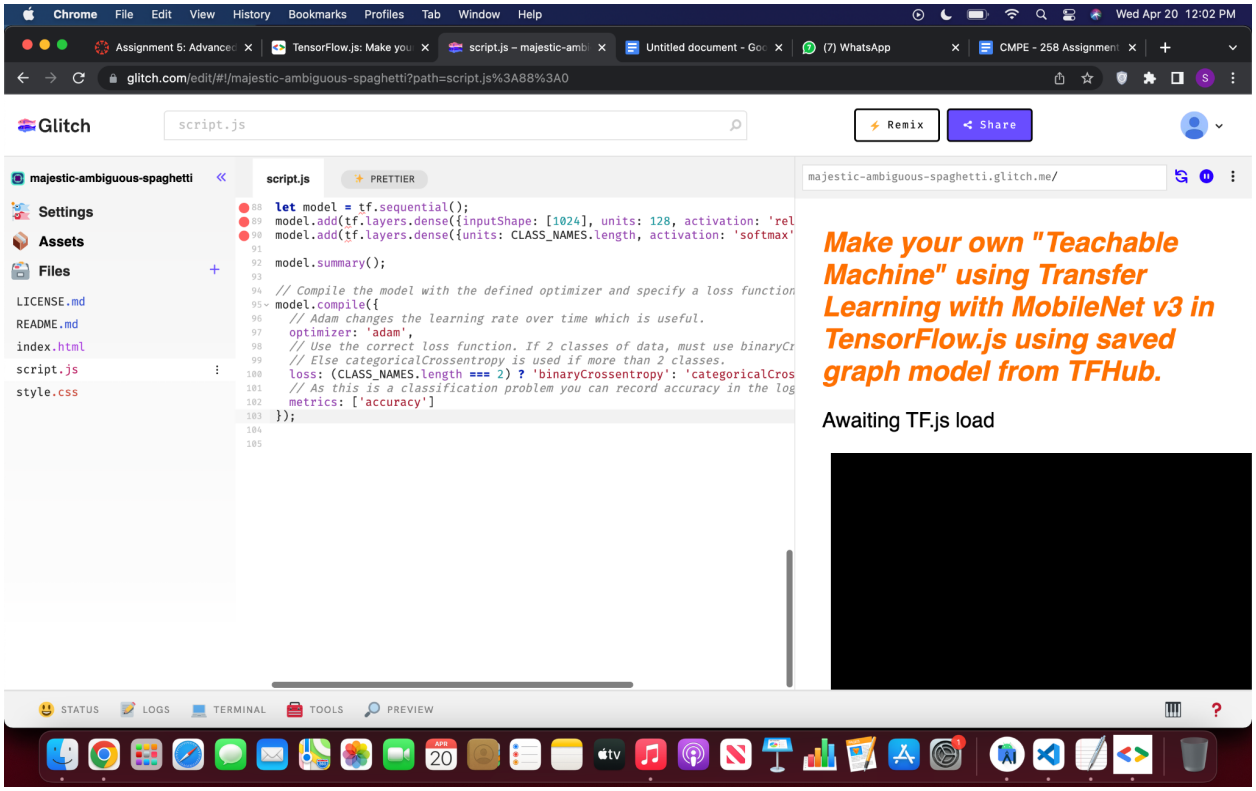
- define a new function called loadMobileNetFeatureModel



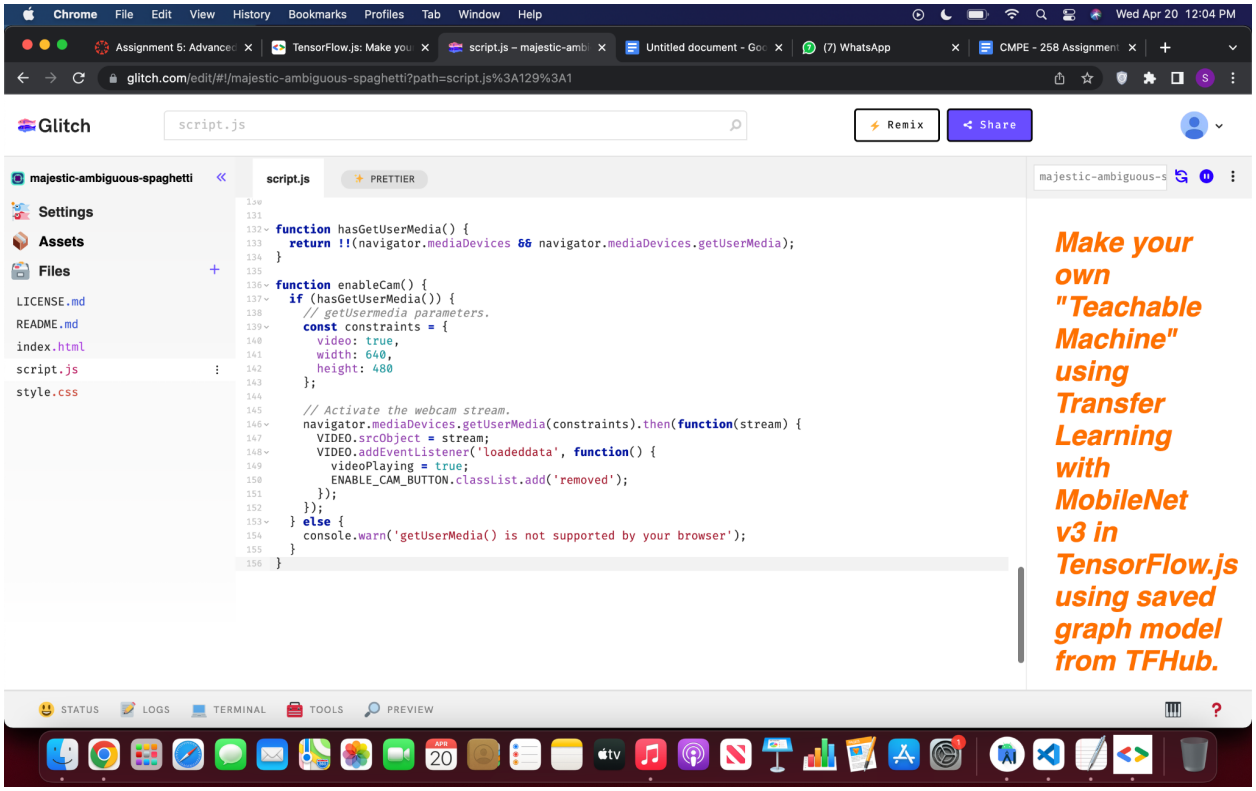
- status text change from "Awaiting TF.js load" to become "MobileNet v3 loaded successfully!"



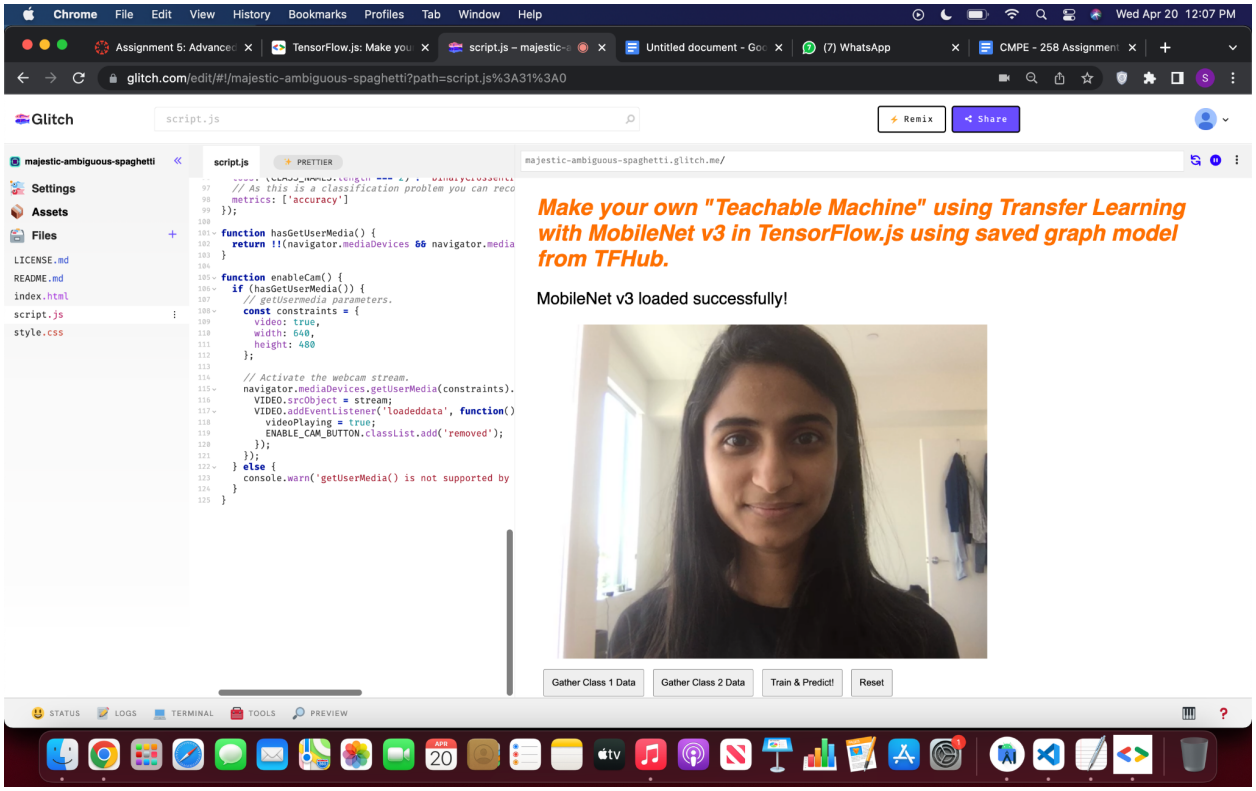
- Define new Model head



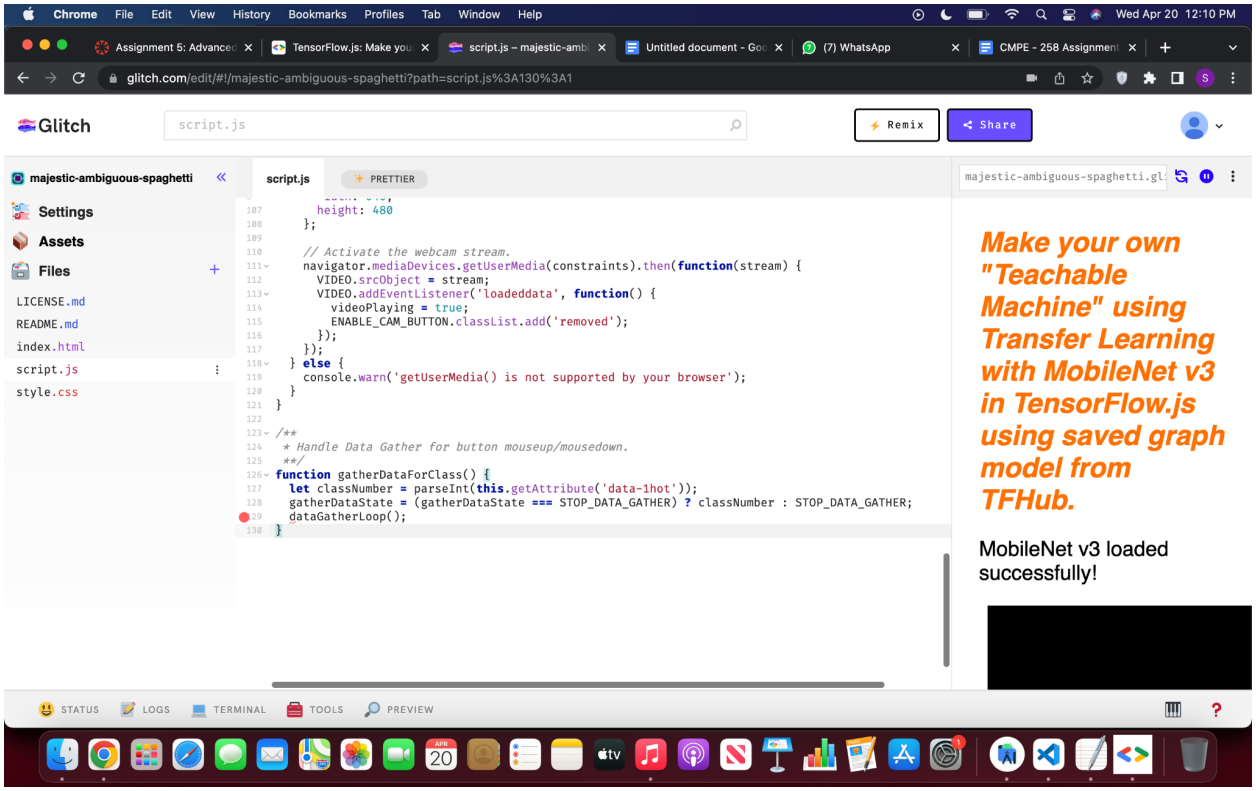
- Enable the webcam



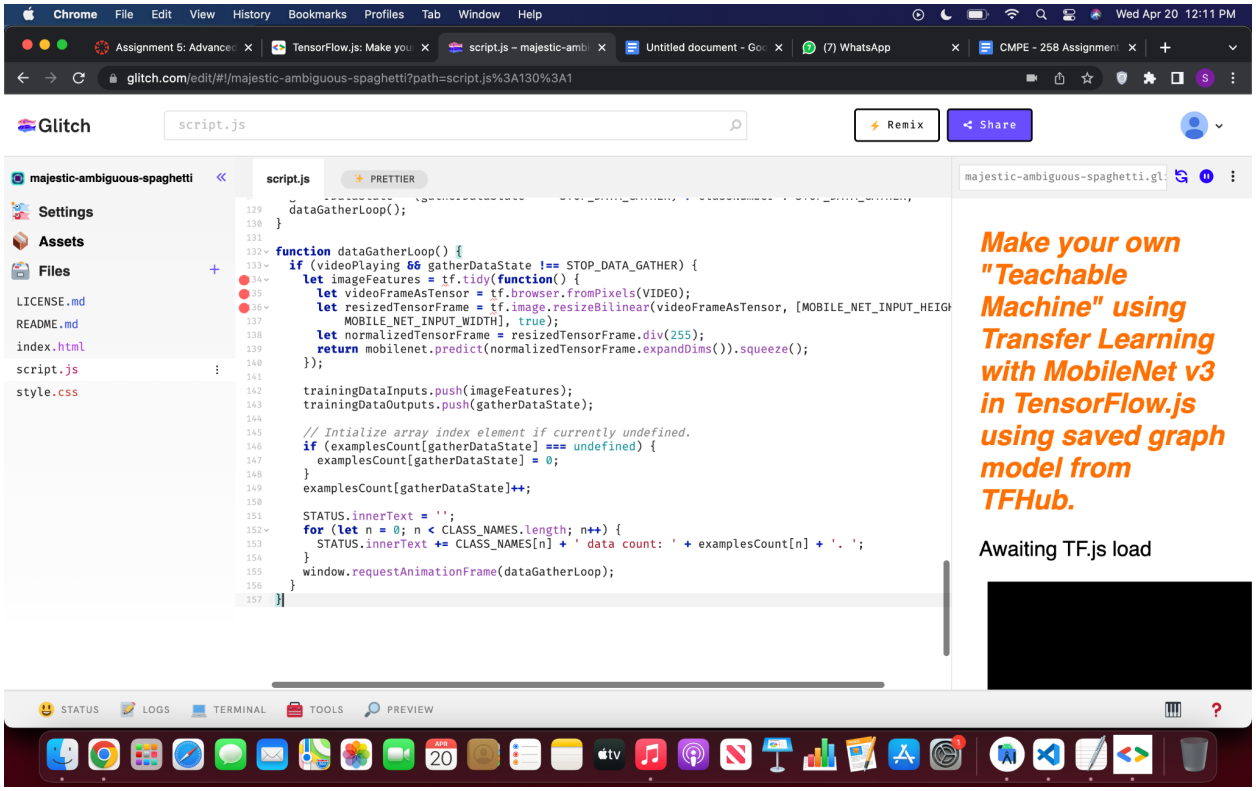
- Checking the preview of webcam



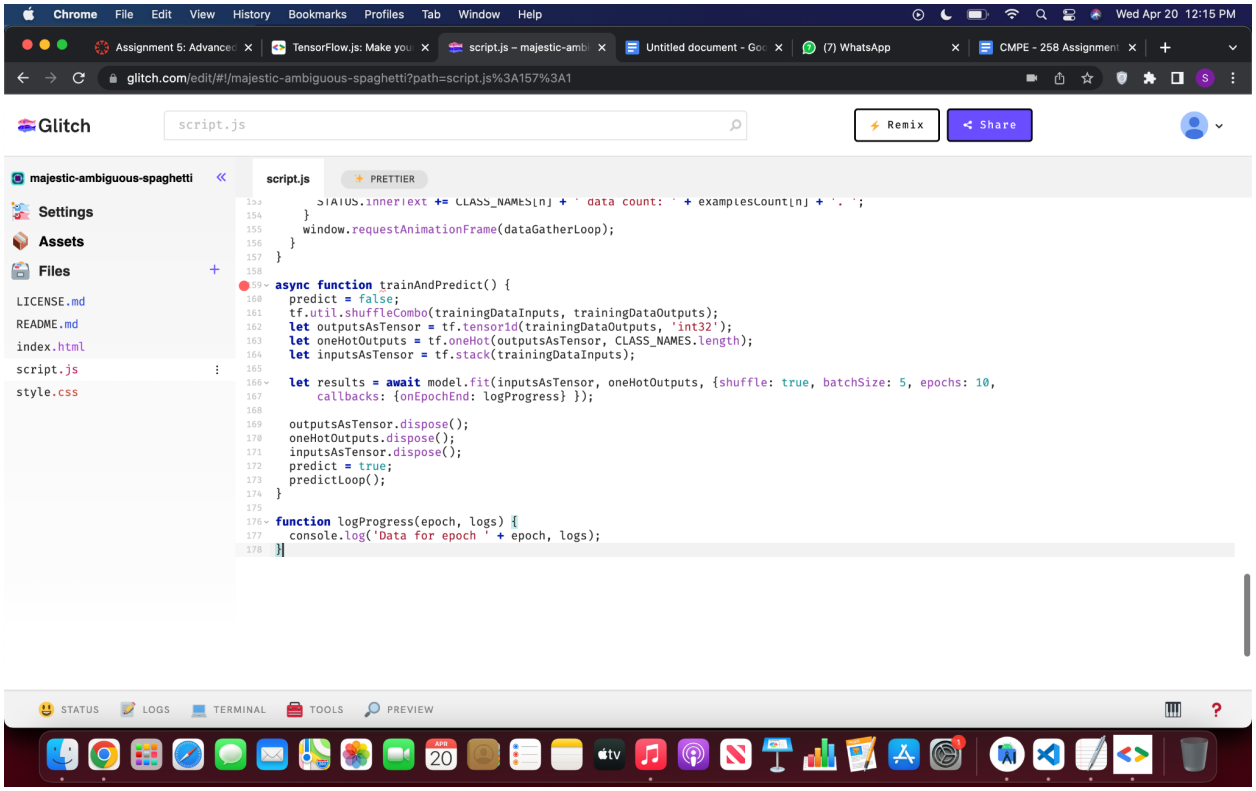
- Data Collection button event handler



- Data Collection

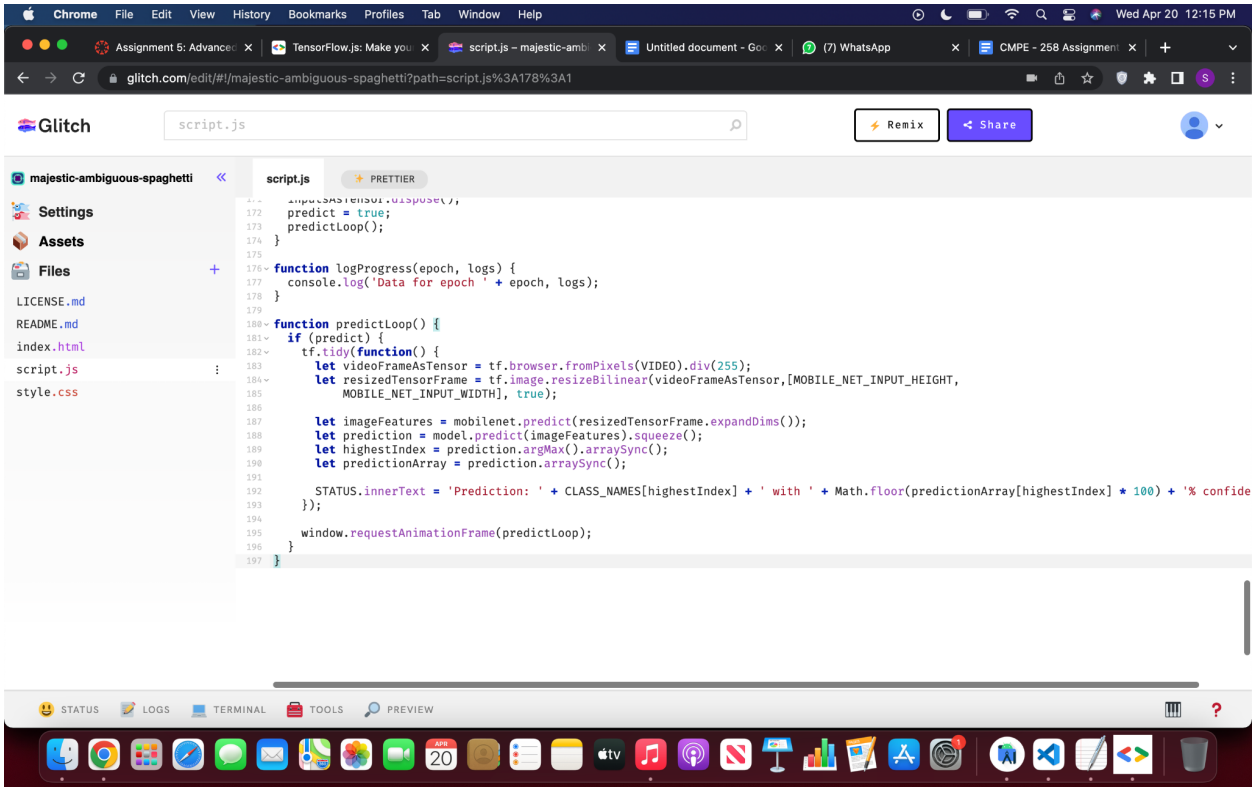


- Train and Predict



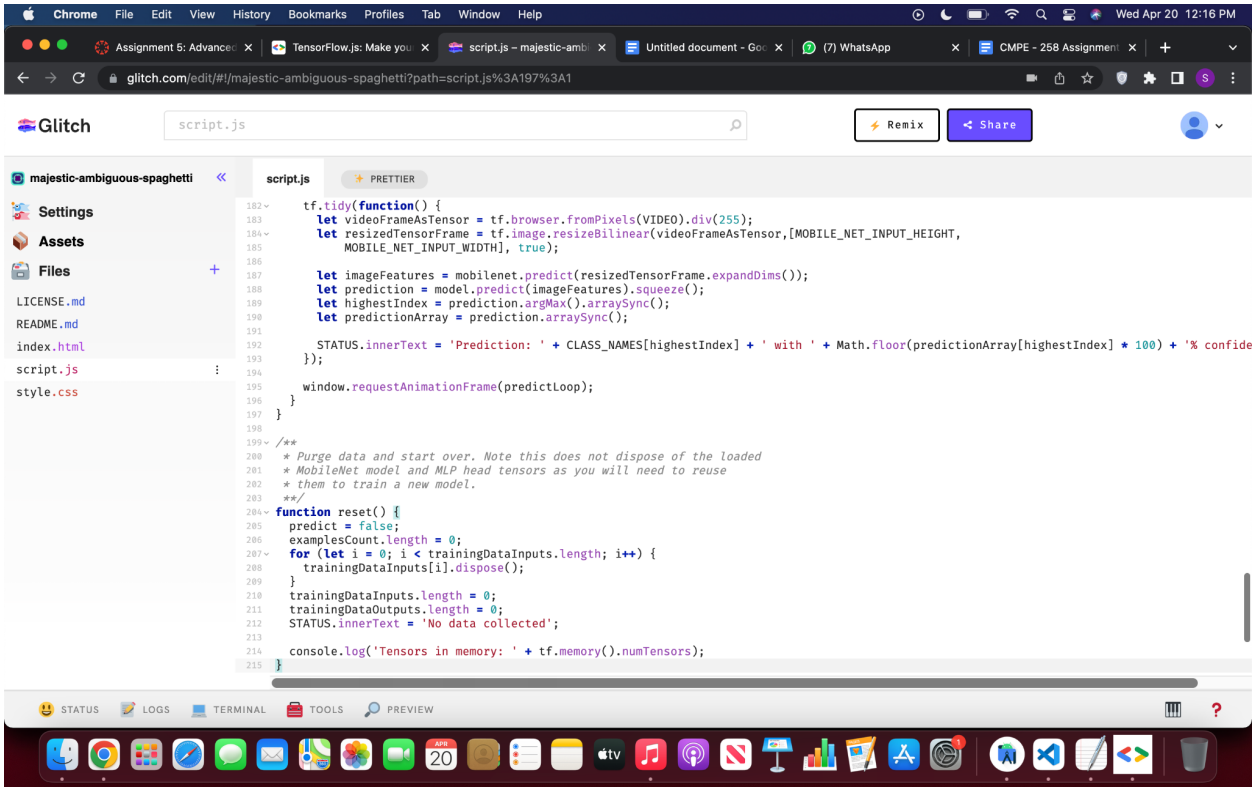
```
async function trainAndPredict() {
  predict = false;
  tf.util.shuffleCombo(trainingDataInputs, trainingDataOutputs);
  let outputsAsTensor = tf.tensor1d(trainingDataOutputs, 'int32');
  let oneHotOutputs = tf.oneHot(outputsAsTensor, CLASS_NAMES.length);
  let inputsAsTensor = tf.stack(trainingDataInputs);

  let results = await model.fit(inputsAsTensor, oneHotOutputs, {shuffle: true, batchSize: 5, epochs: 10,
      callbacks: {onEpochEnd: logProgress} });

  outputsAsTensor.dispose();
  oneHotOutputs.dispose();
  inputsAsTensor.dispose();
  predict = true;
  predictLoop();
}

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

- Predict Loop



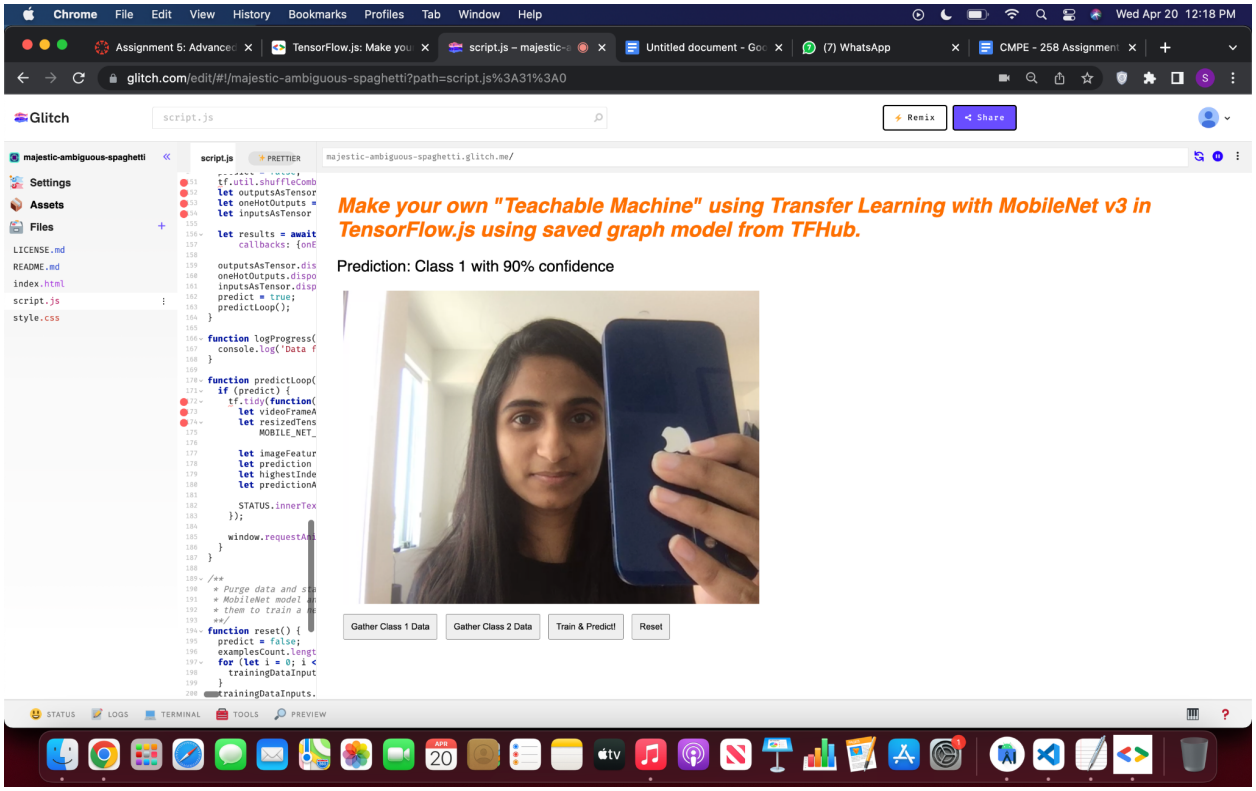
```
  predict = true;
  predictLoop();
}

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

function predictLoop() {
  if (predict) {
    tf.tidy(function() {
      let videoFrameAsTensor = tf.browser.fromPixels(VIDEO).div(255);
      let resizedTensorFrame = tf.image.resizeBilinear(videoFrameAsTensor,[MOBILE_NET_INPUT_HEIGHT,
          MOBILE_NET_INPUT_WIDTH], true);

      let imageFeatures = mobilenet.predict(resizedTensorFrame.expandDims());
      let prediction = model.predict(imageFeatures).squeeze();
      let highestIndex = prediction.argMax().arraySync();
      let predictionArray = prediction.arraySync();

      STATUS.innerText = 'Prediction: ' + CLASS_NAMES[highestIndex] + ' with ' + Math.floor(predictionArray[highestIndex] * 100) + '% confide
    });

    window.requestAnimationFrame(predictLoop);
  }
}
```

- Implement the reset button



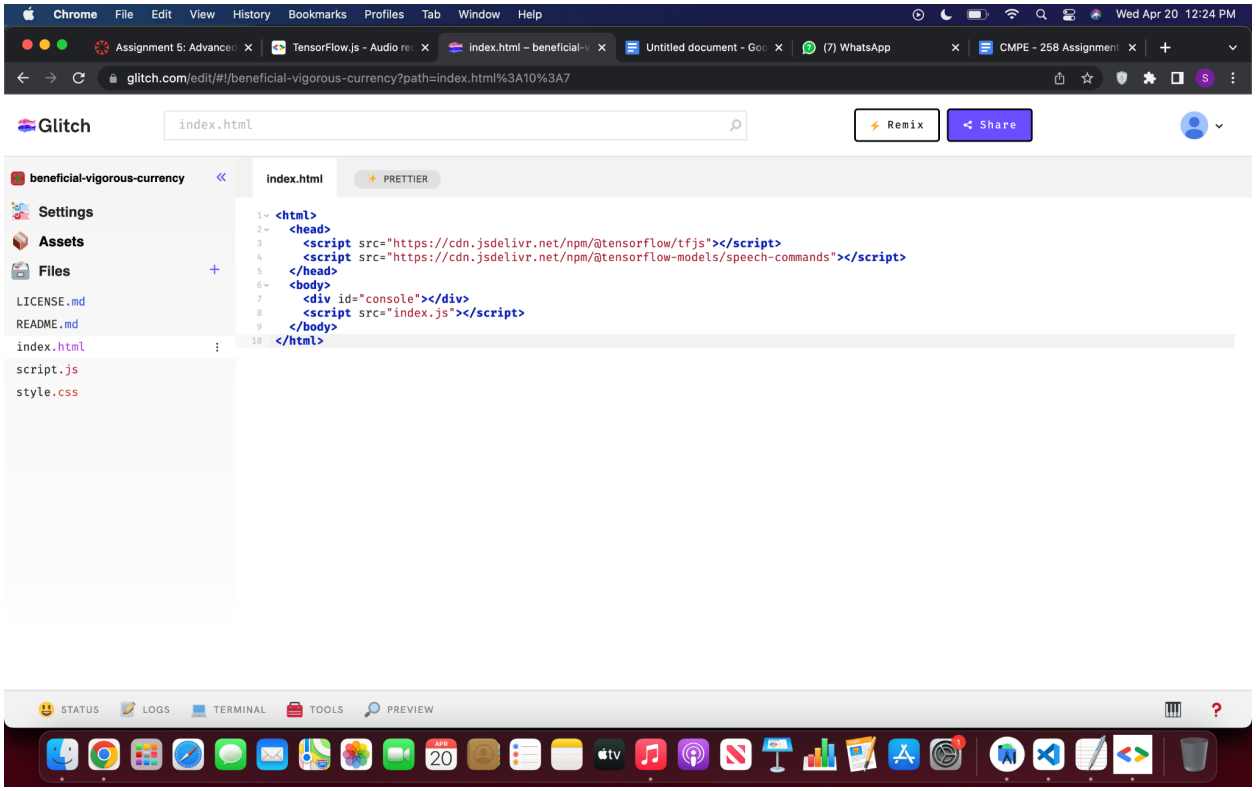
```
function reset() {
  predict = false;
  examplesCount.length = 0;
  for (let i = 0; i < trainingDataInputs.length; i++) {
    trainingDataInputs[i].dispose();
  }
  trainingDataInputs.length = 0;
  trainingDataOutputs.length = 0;
  STATUS.innerText = 'No data collected';

  console.log('Tensors in memory: ' + tf.memory().numTensors);
}
```

- Prediction Results

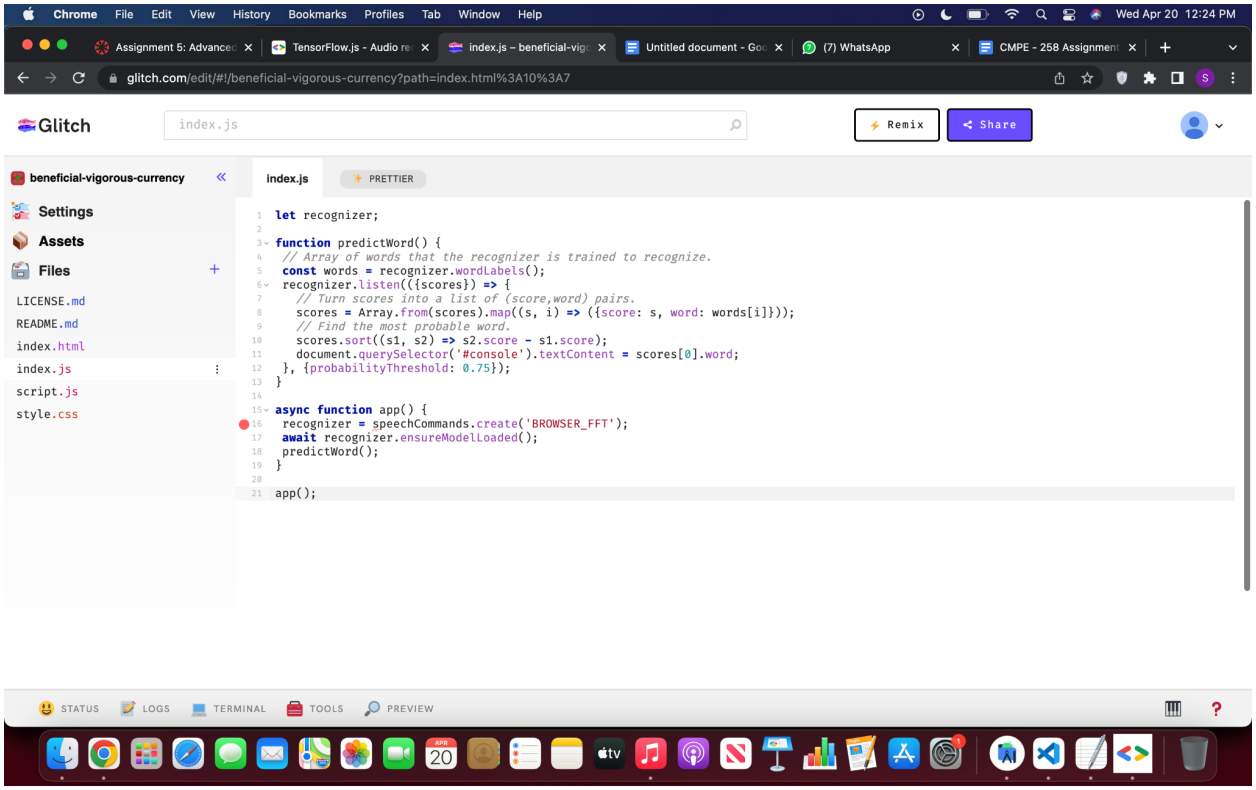


D) Webapp and audio training on device : TensorFlow.js - Audio recognition using transfer learning

- Load tensorflow.js and audio model



- Create and index.js file and add the content to predict in real time



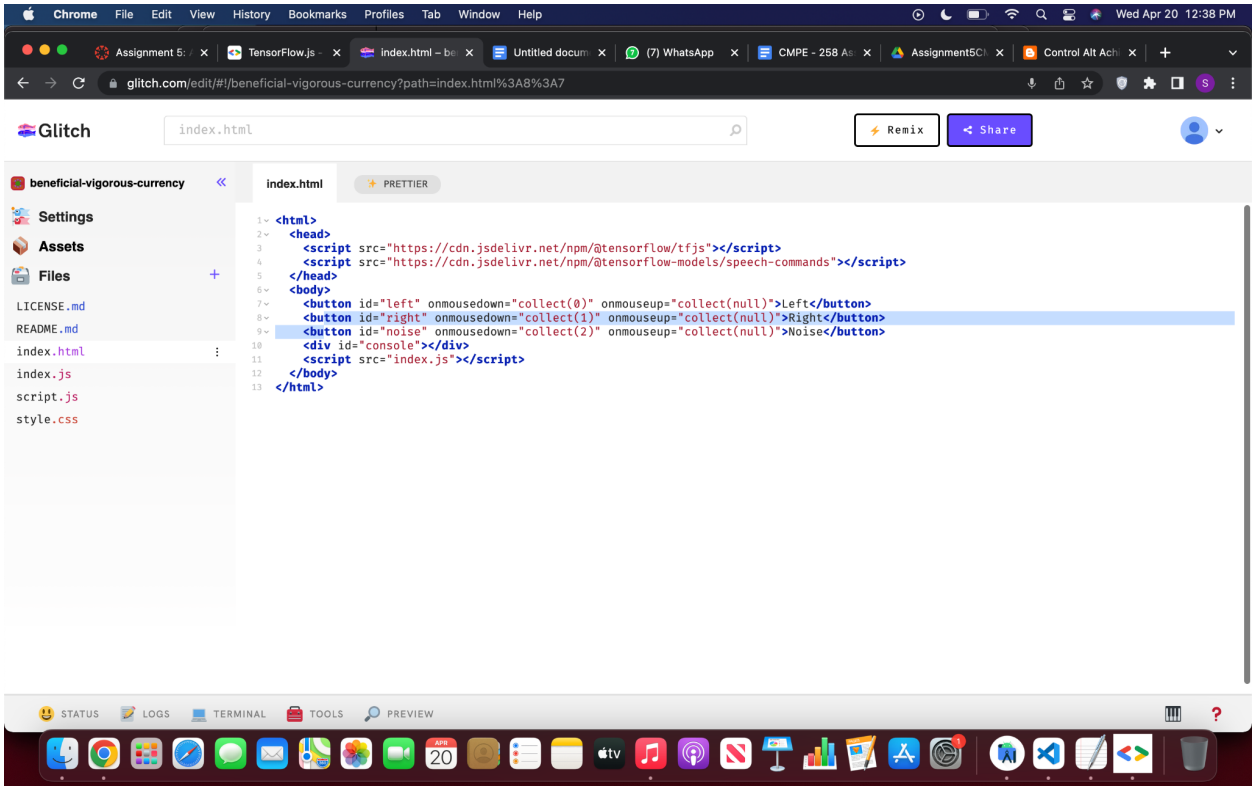
- Audio prediction results

Saying the words are giving real time predictions

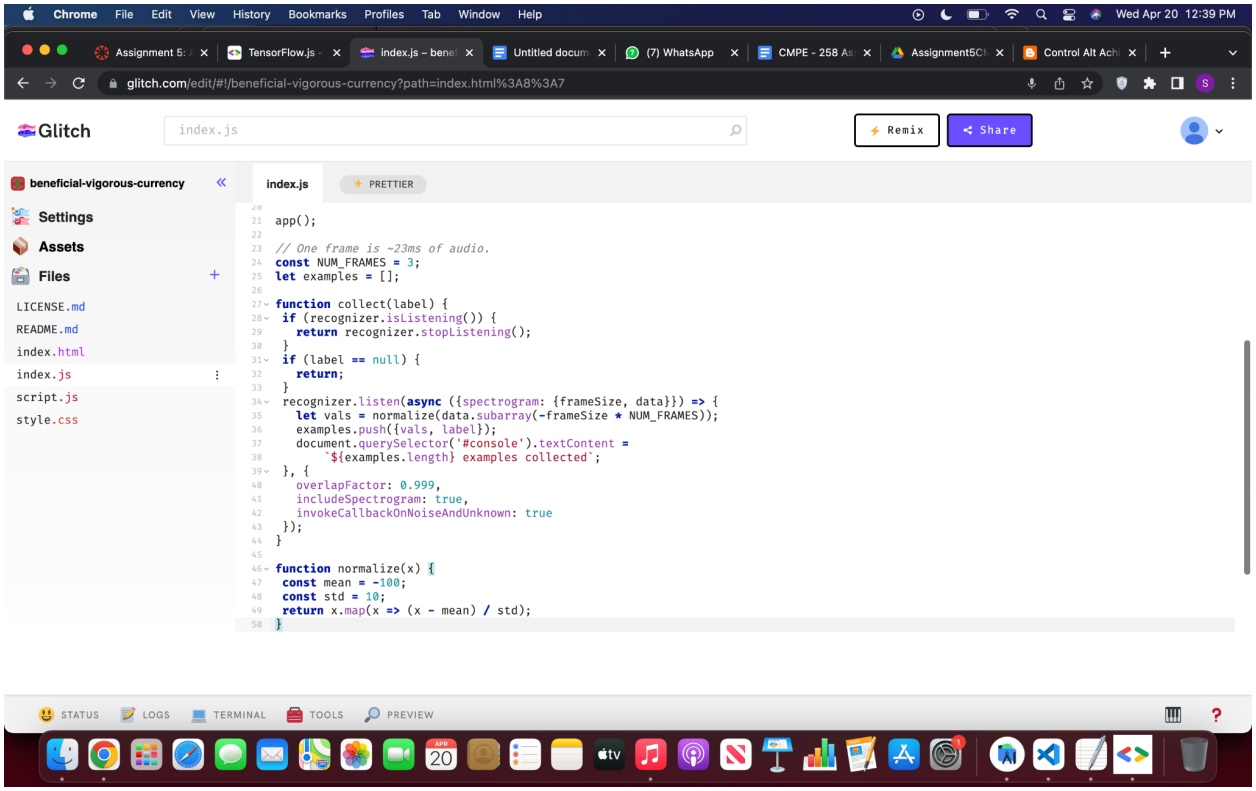
[Drive link to the prediction results]

- Collect Data

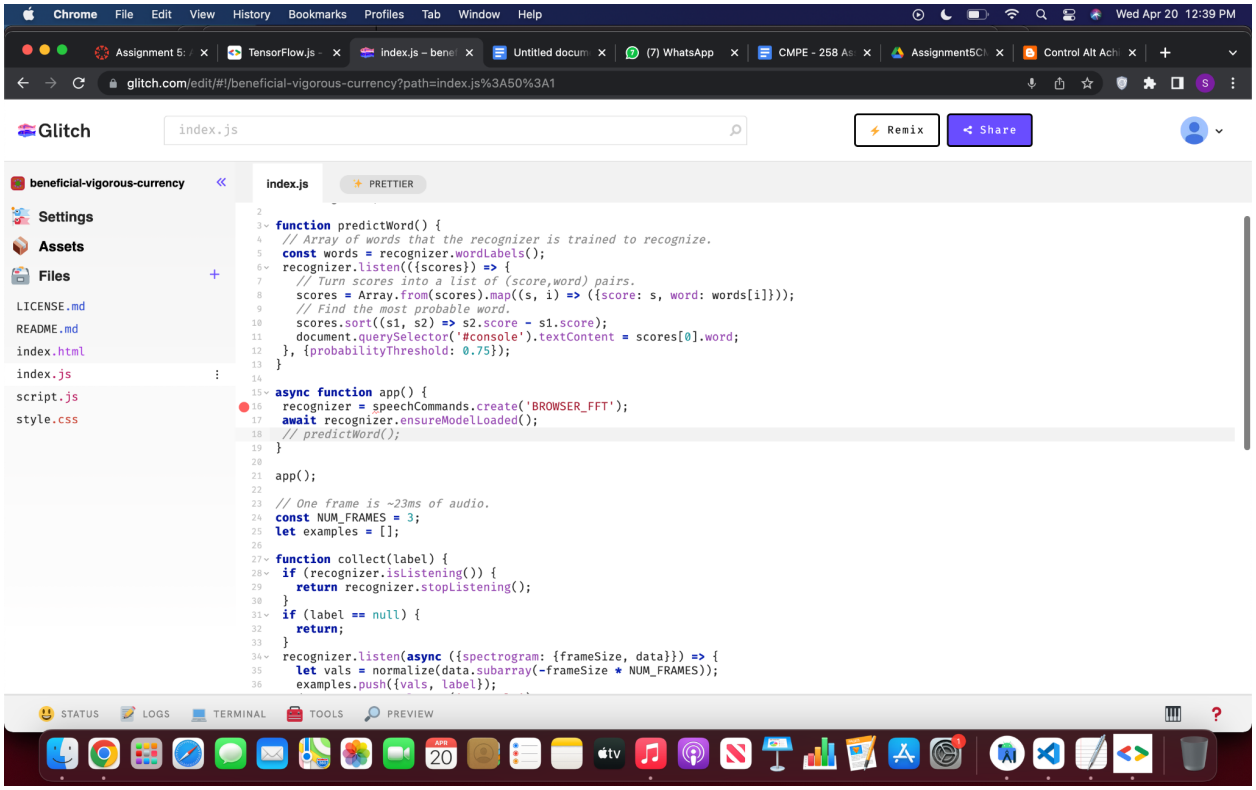
Update index.html



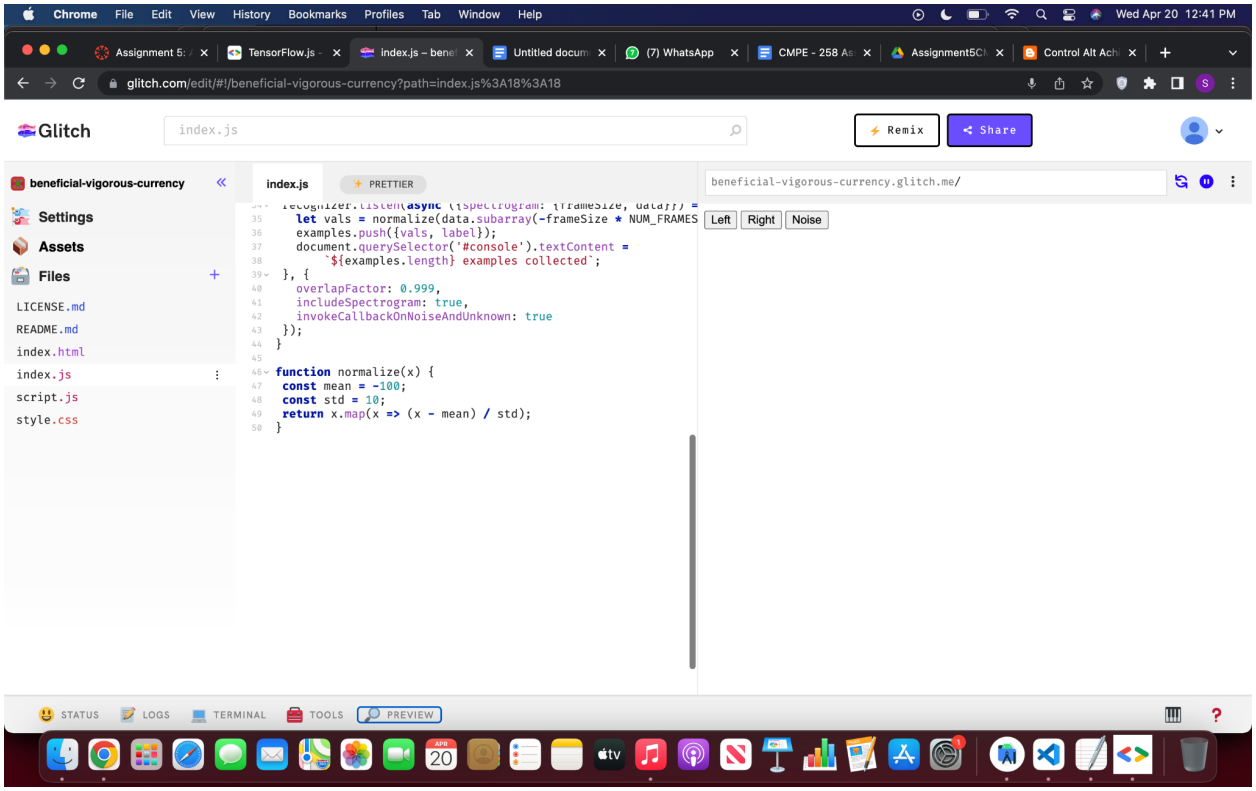
Update index.js



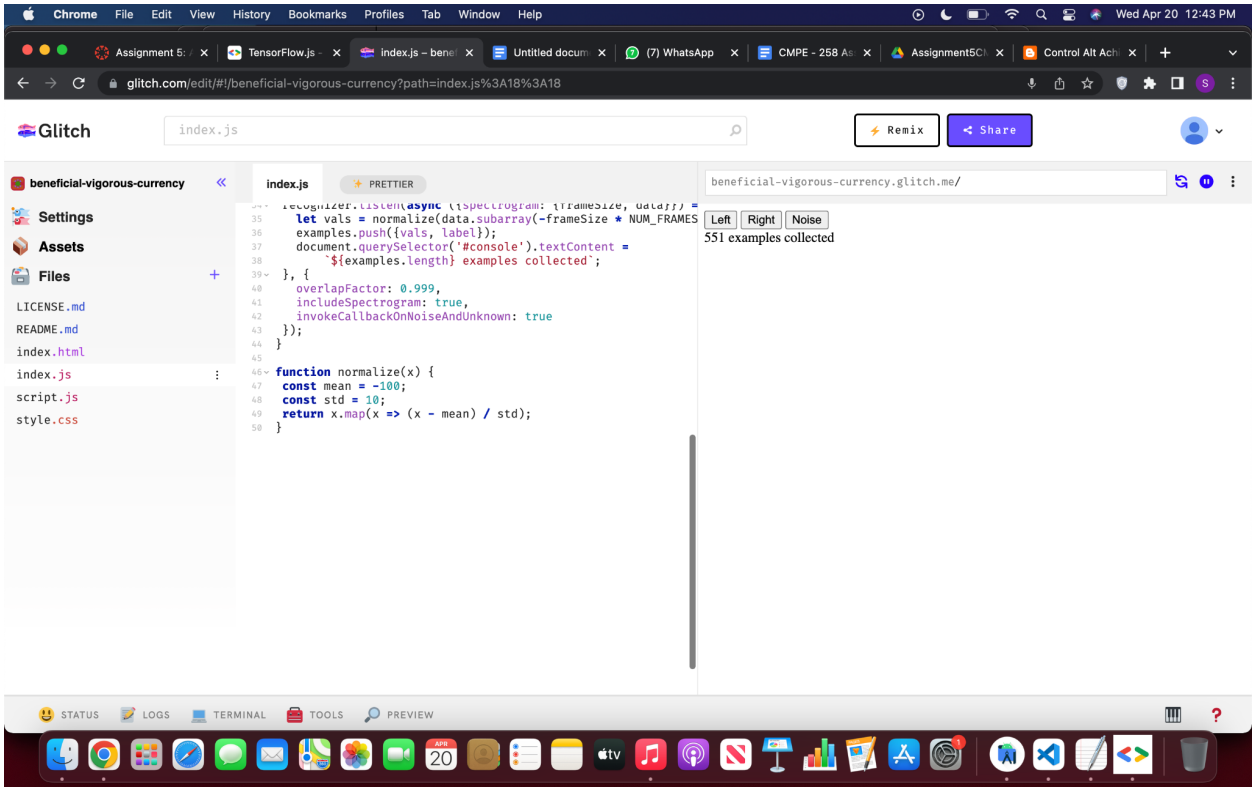
Remove predictWord() call



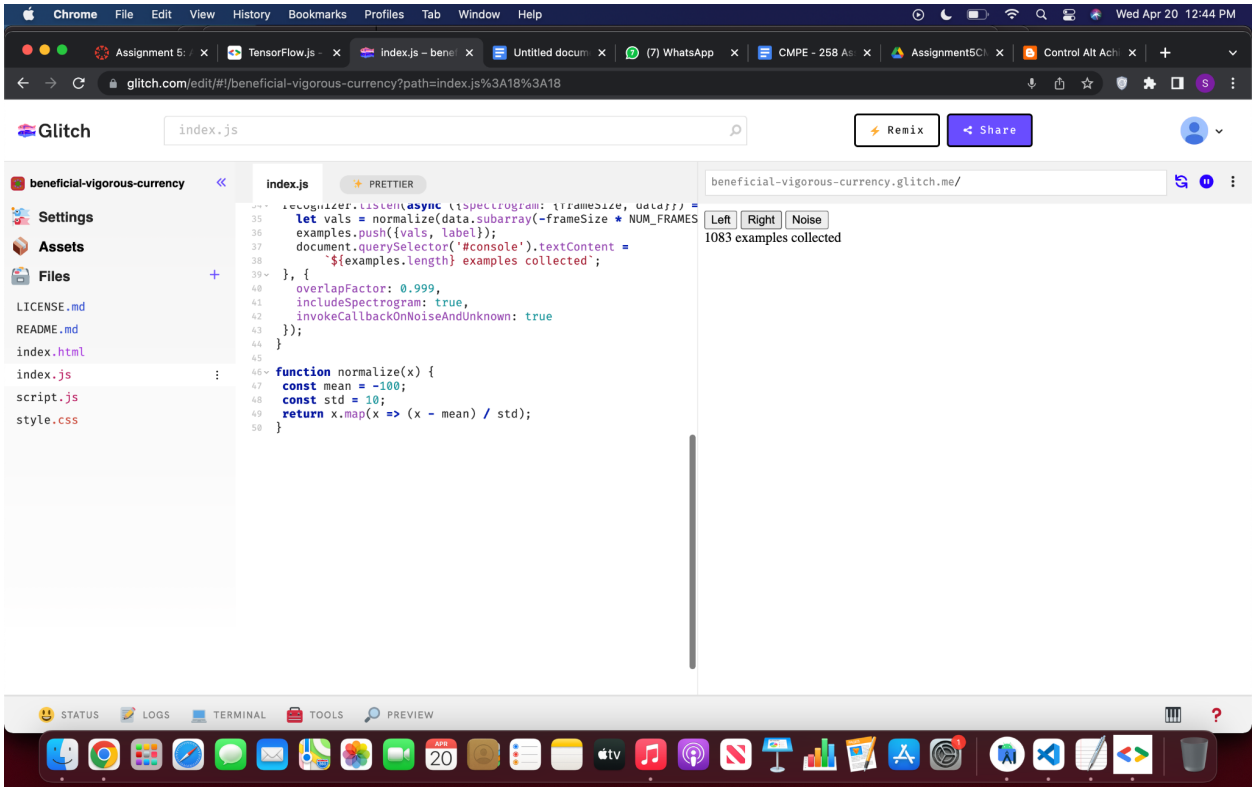
- Test Data Collection



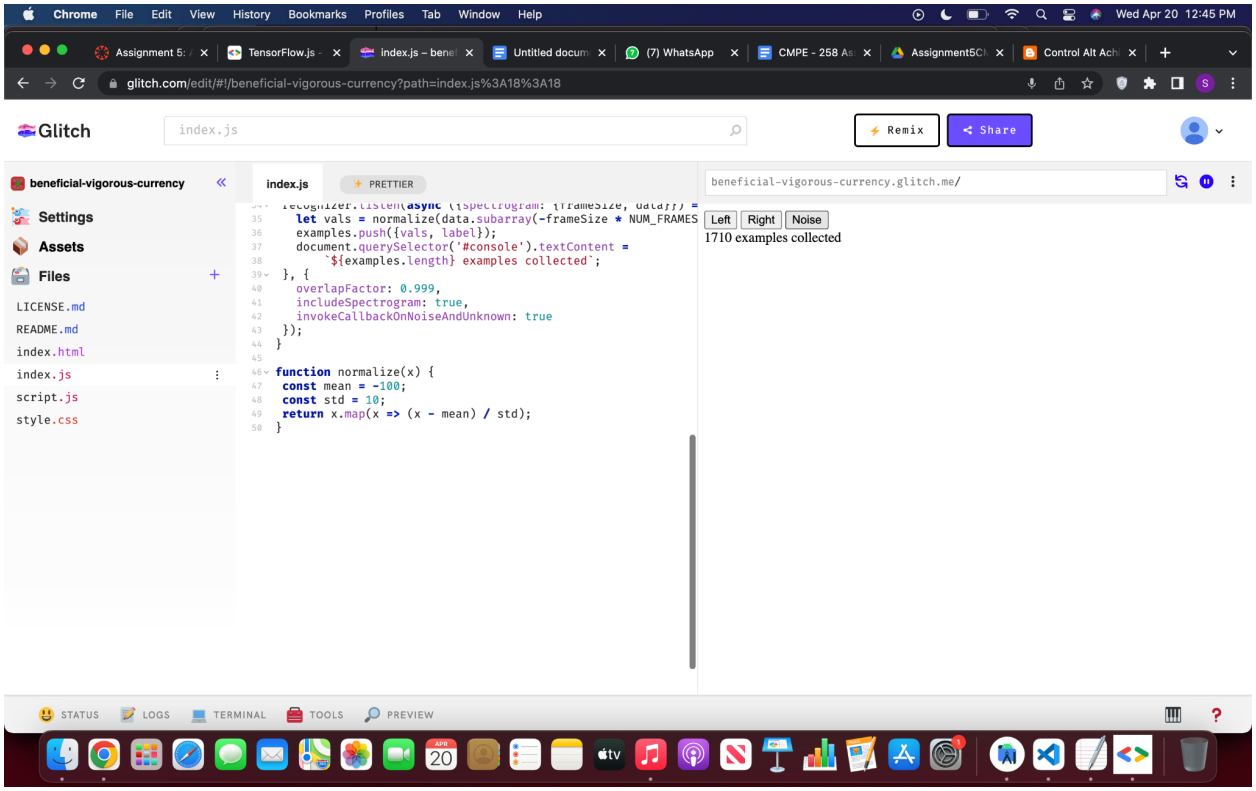
- For Left, I have trained with clap sound



- For right, I have trained with whistle sound

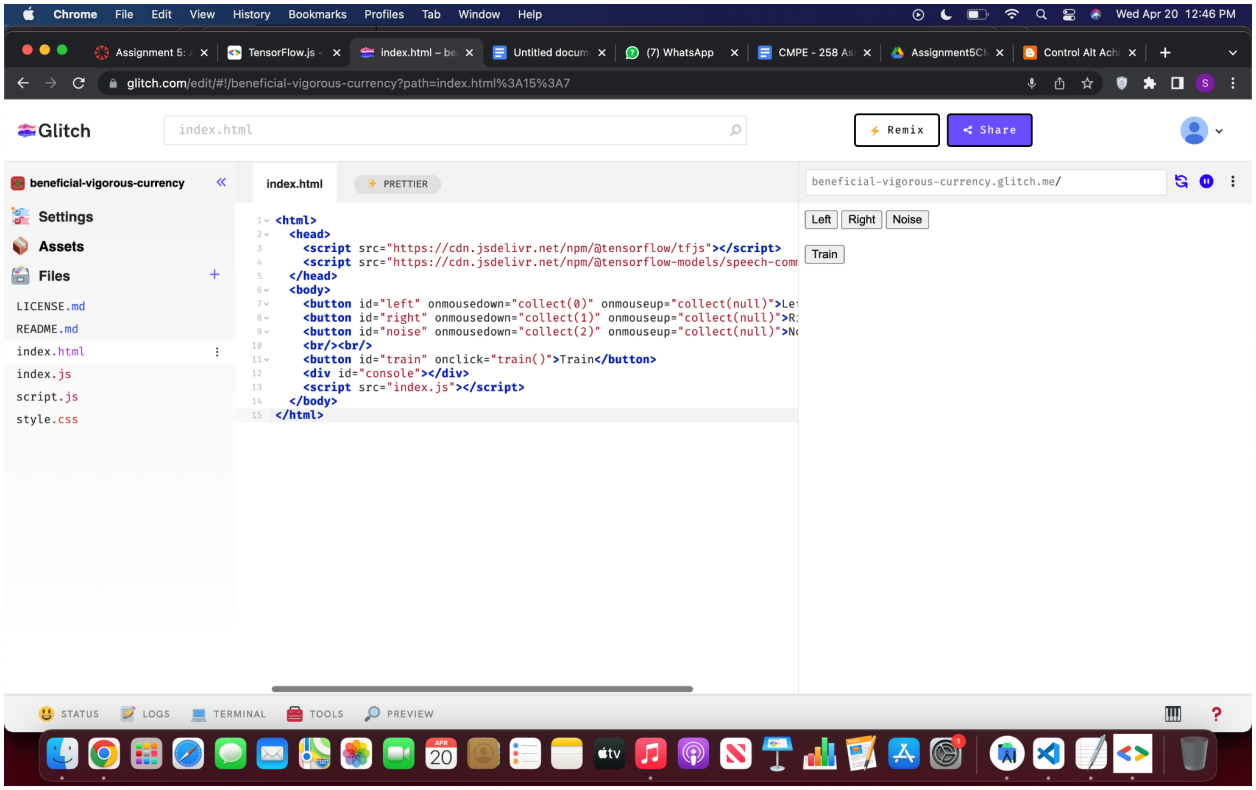


- For noise, I have trained with talking

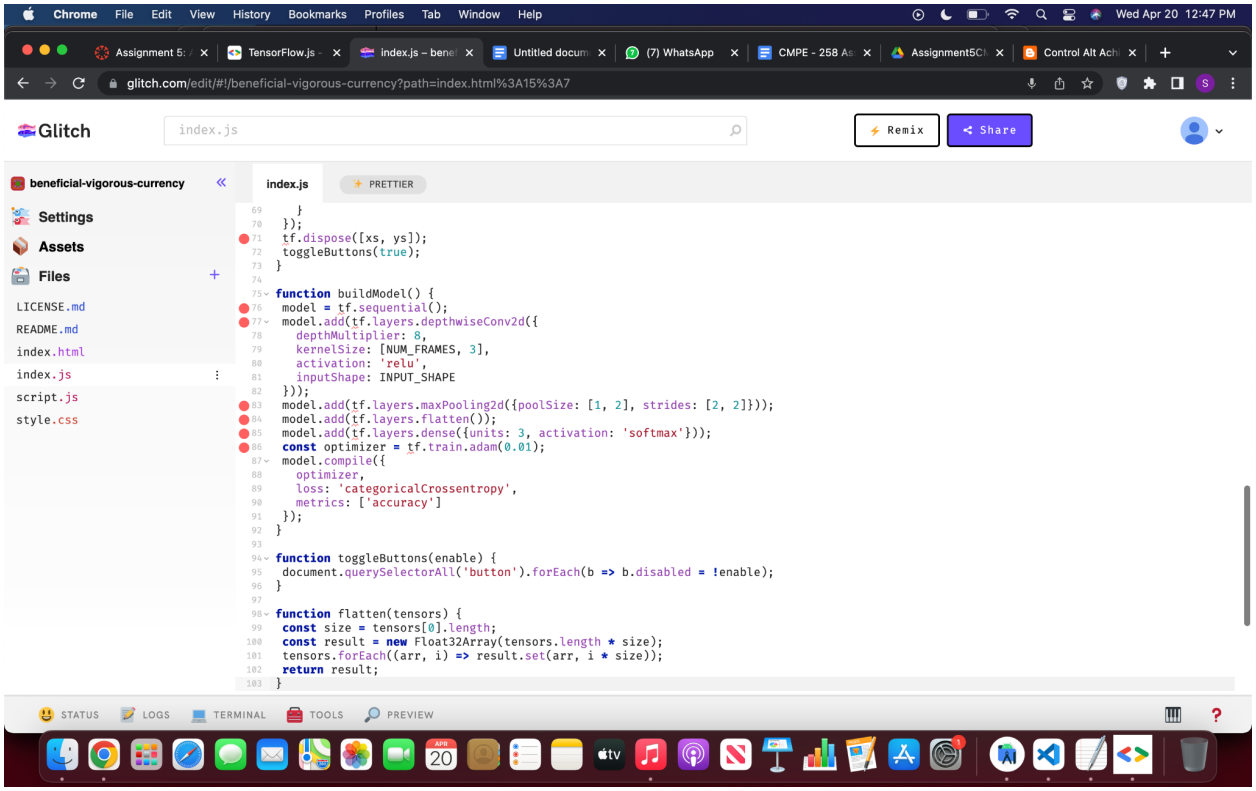


- Train a model

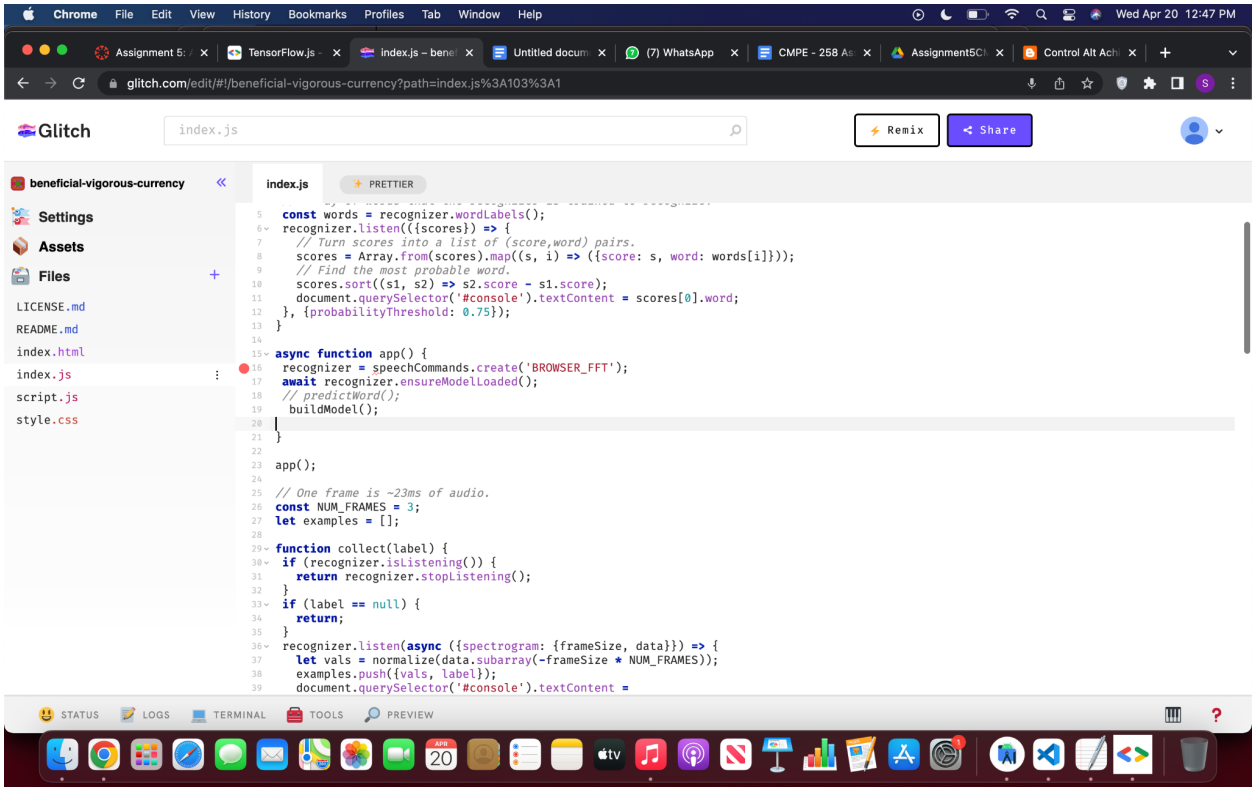
Add train button right after noise button



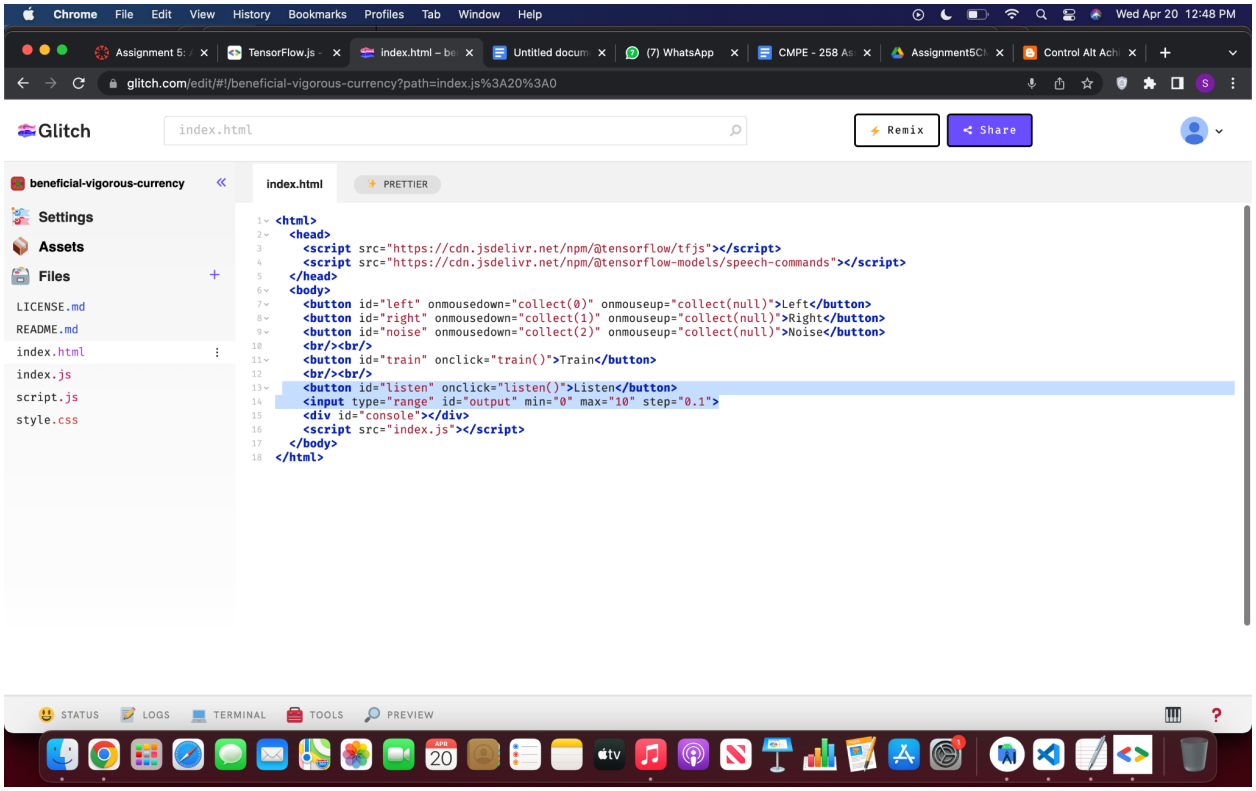
Update index.js



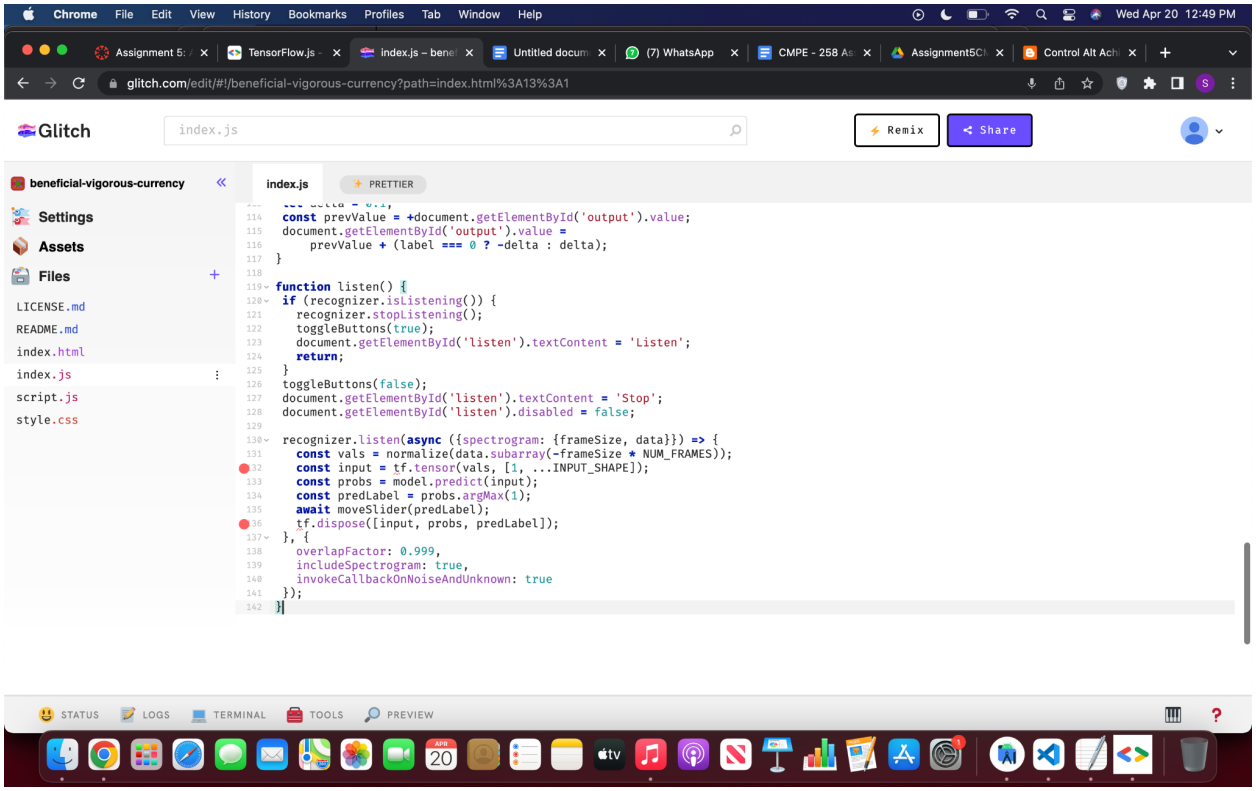
Call buildModel() when app loads



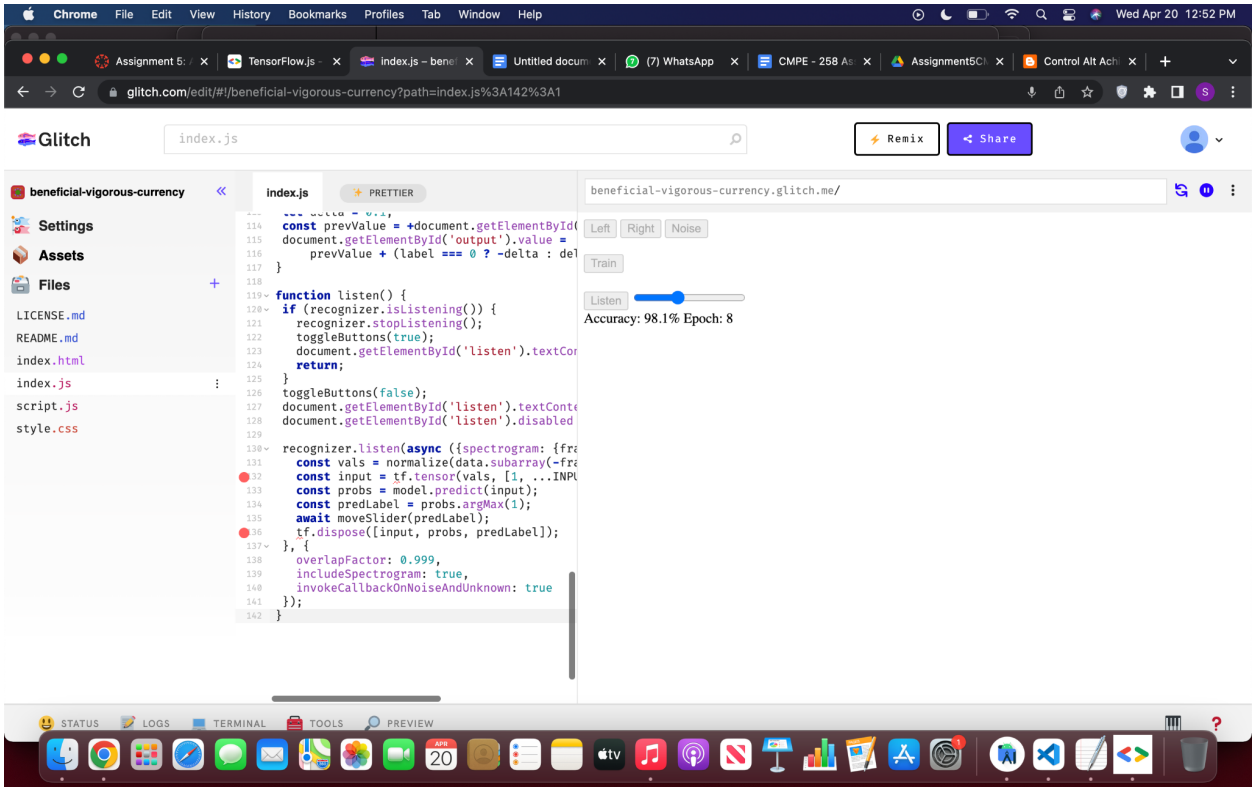
- Update slider in real time



Update index.js

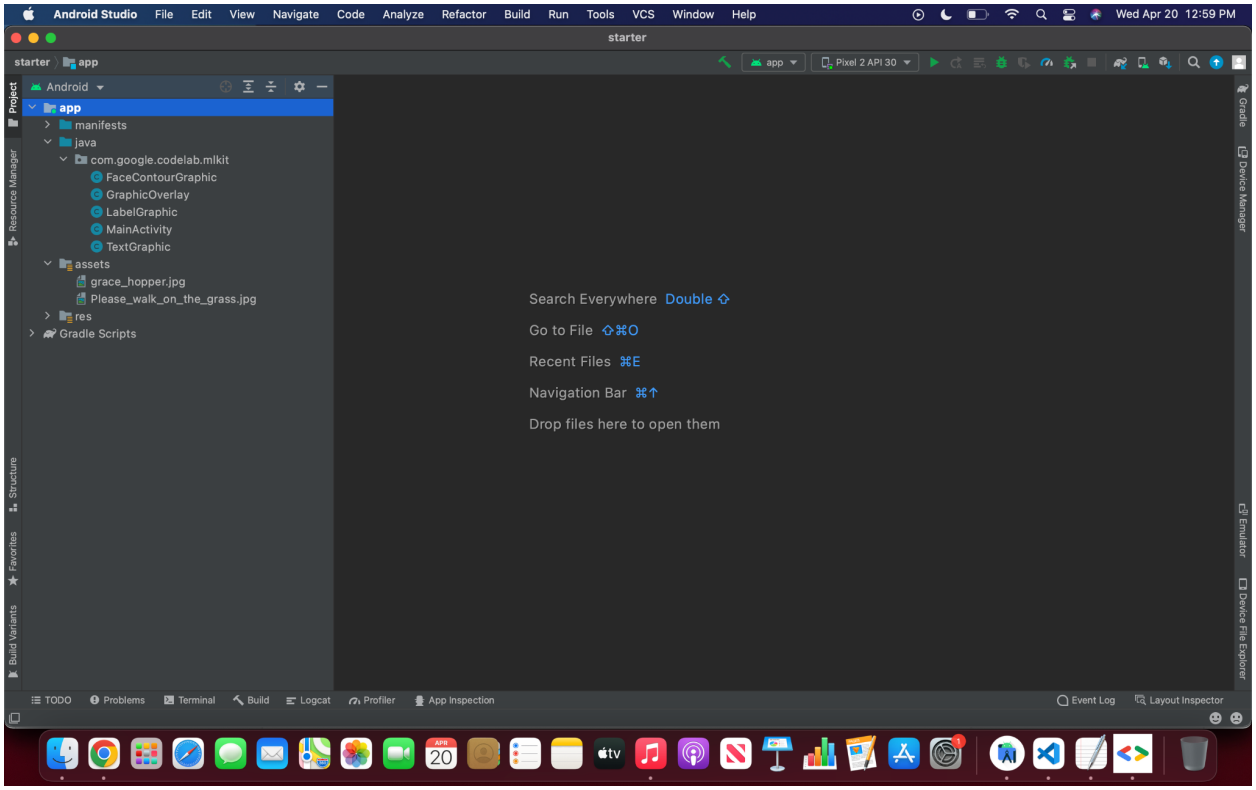


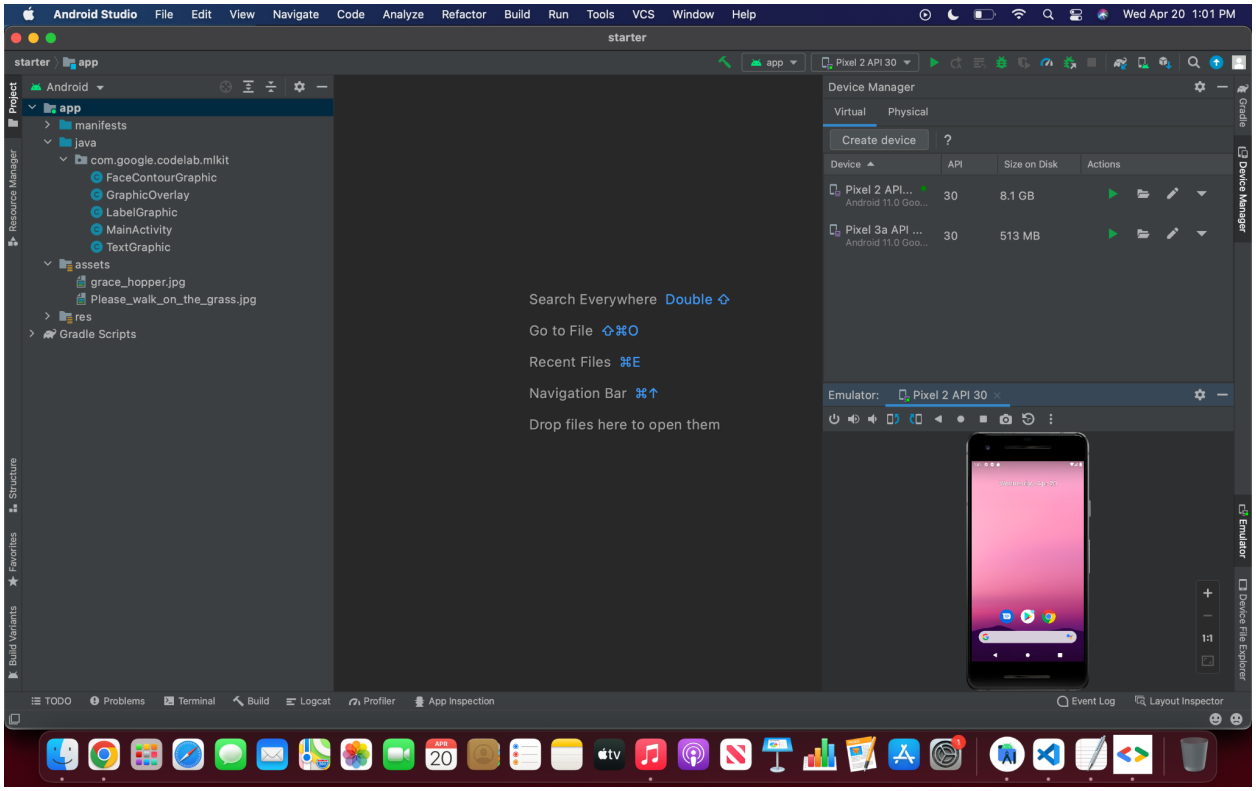
- Testing the final result



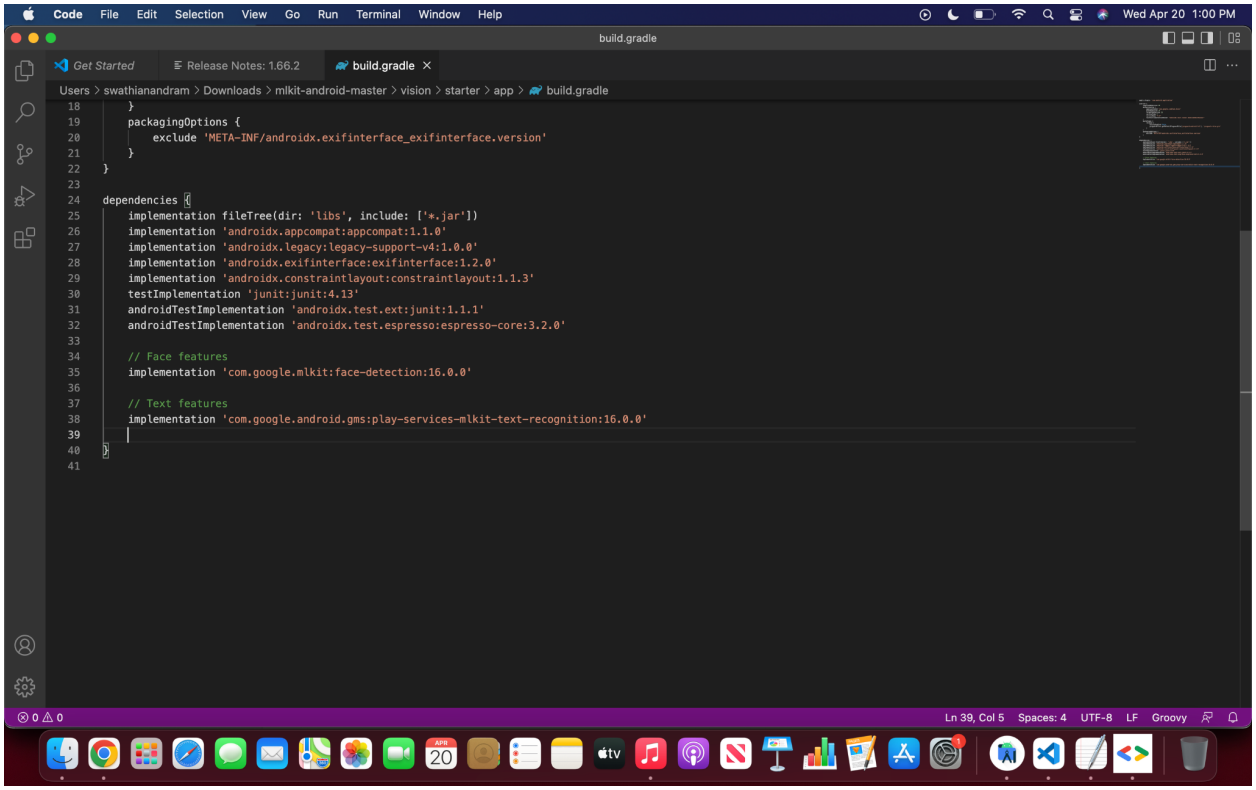
E) Using out of box sdk to do ML ondevice : Use ML Kit to perform Recognize text and facial features with ML Kit: Android and Recognize, Identify Language and Translate text with ML Kit and CameraX: Android

- Setting up the project on Android Studio

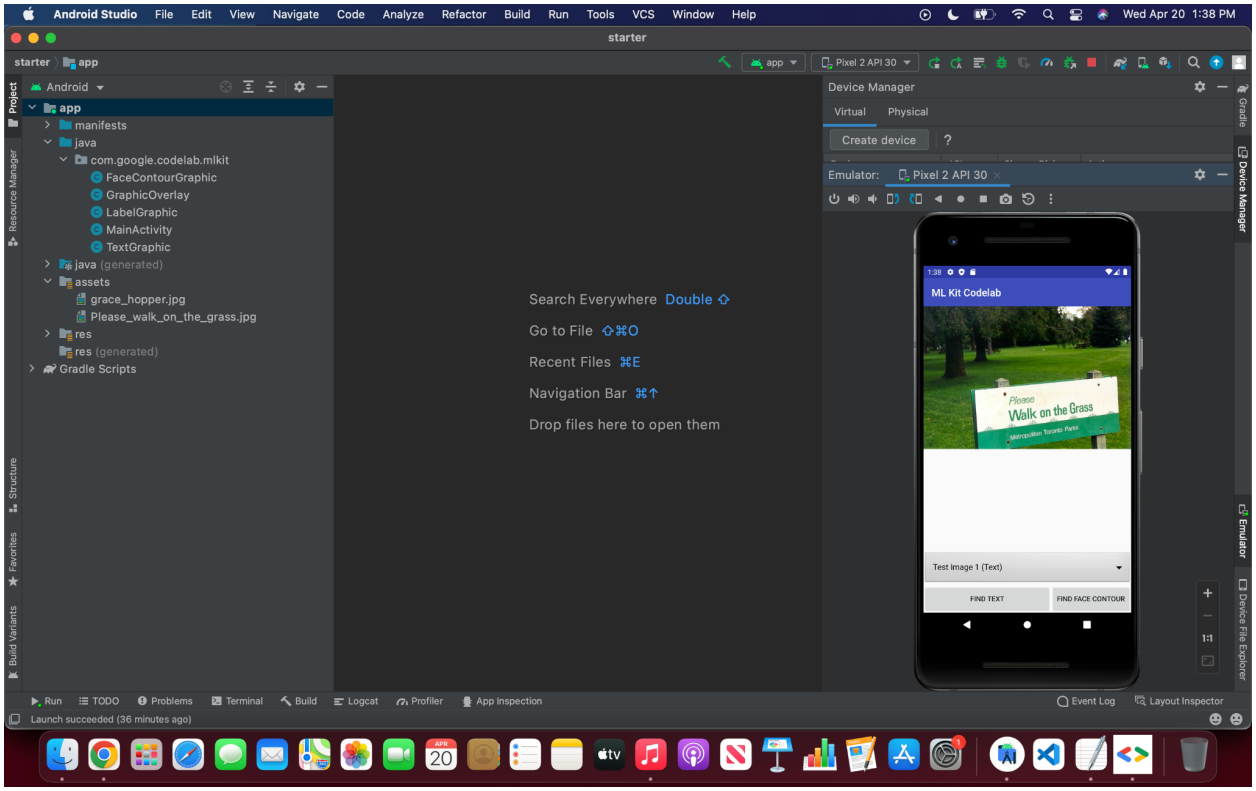




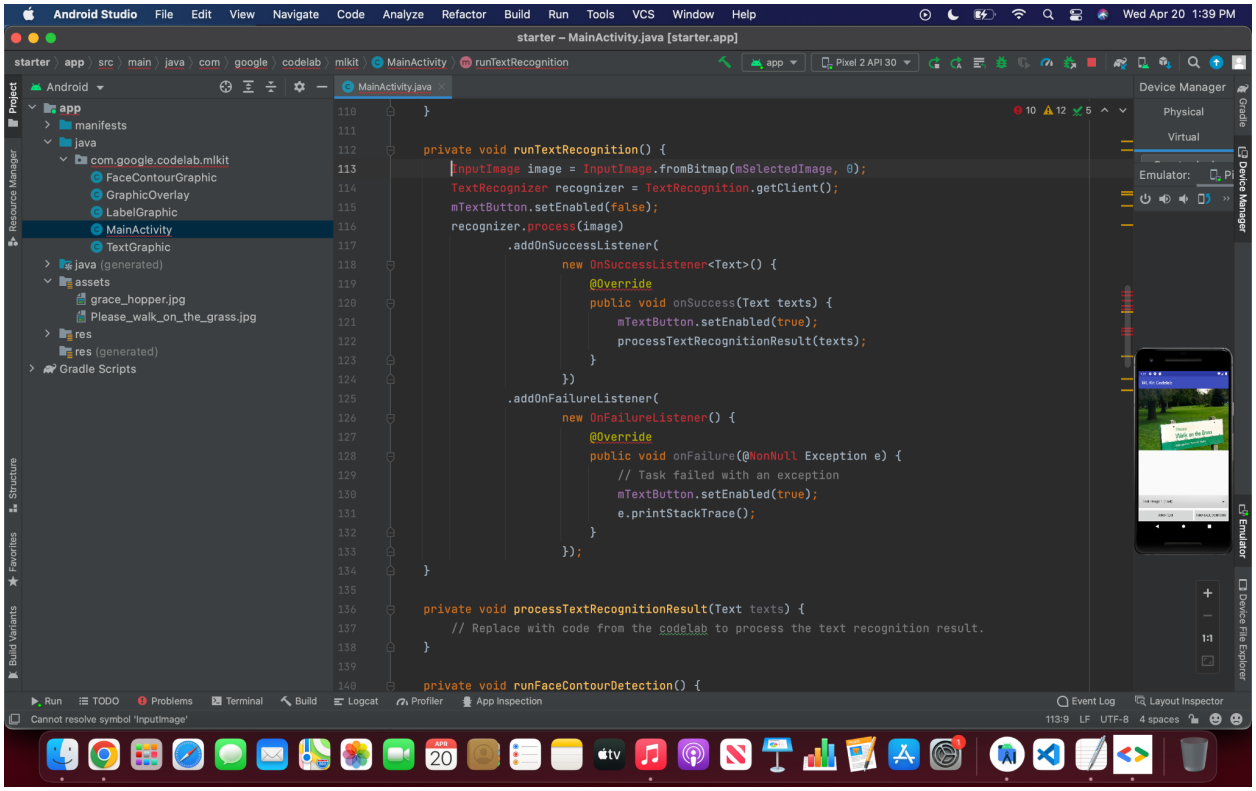
- Checking necessary dependencies



- Run starter app

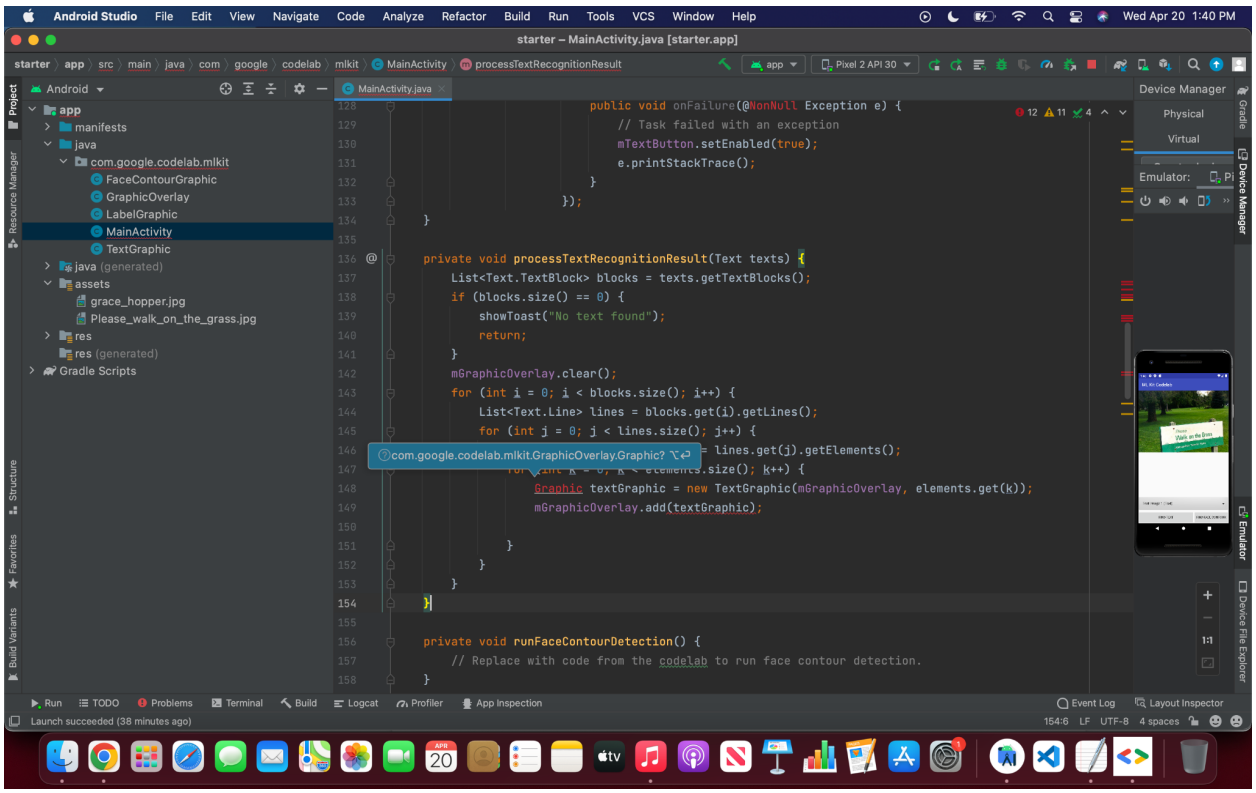


- Update runTextRecognition function



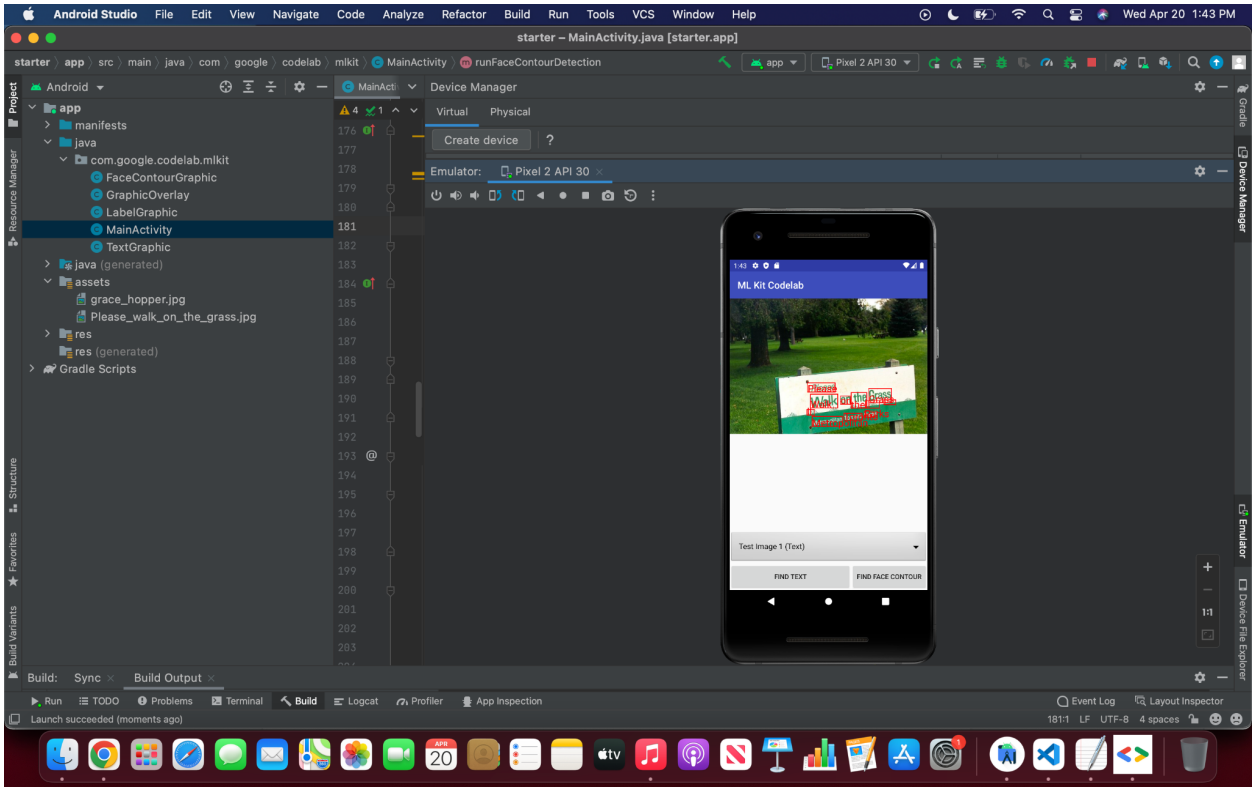

```
}

private void runTextRecognition() {
    InputImage image = InputImage.fromBitmap(mSelectedImage, 0);
    TextRecognizer recognizer = TextRecognition.getClient();
    mTextButton.setEnabled(false);
    recognizer.process(image)
            .addOnSuccessListener(
                    new OnSuccessListener<Text>() {
                        @Override
                        public void onSuccess(Text texts) {
                            mTextButton.setEnabled(true);
                            processTextRecognitionResult(texts);
                        }
                    })
            .addOnFailureListener(
                    new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            // Task failed with an exception
                            mTextButton.setEnabled(true);
                            e.printStackTrace();
                        }
                    });
}

private void processTextRecognitionResult(Text texts) {
    // Replace with code from the codelab to process the text recognition result.
}

private void runFaceContourDetection() {
```

- Update processTextRecognitionResult() function

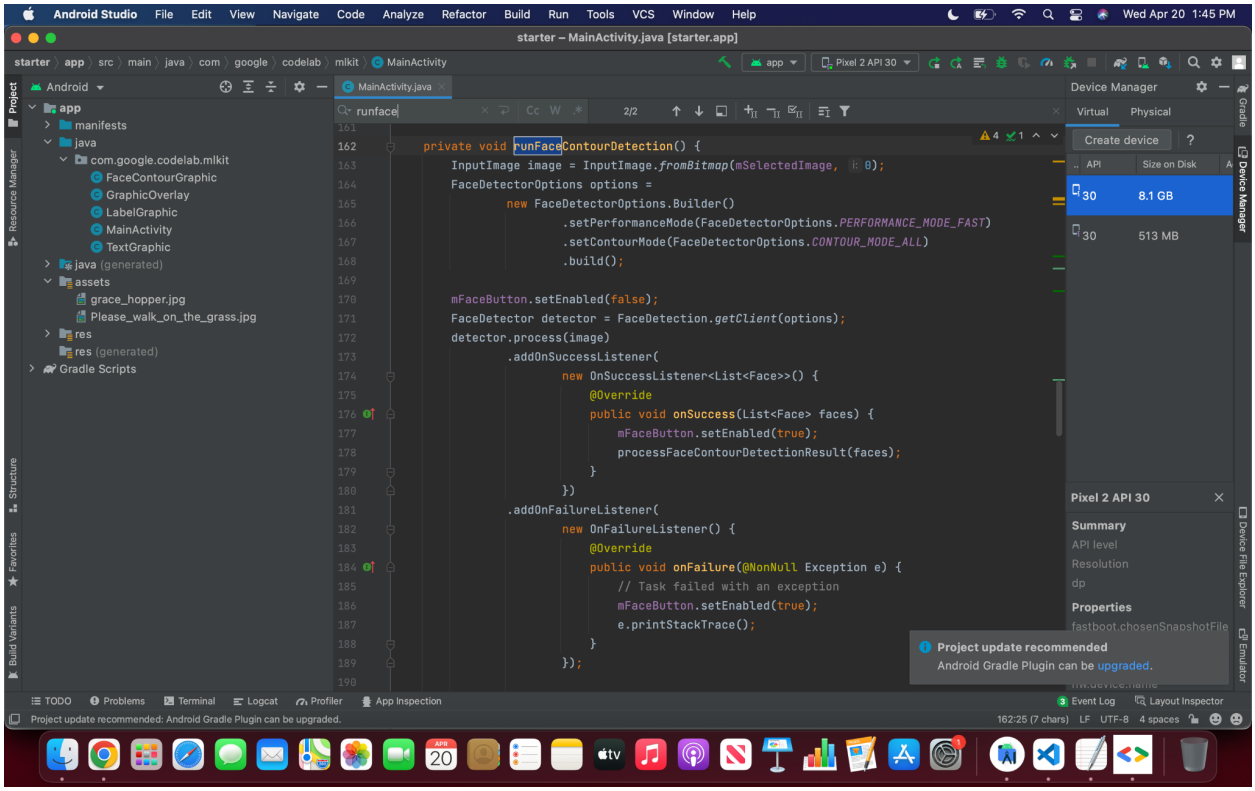



```
                        public void onFailure(@NonNull Exception e) {
                            // Task failed with an exception
                            mTextButton.setEnabled(true);
                            e.printStackTrace();
                        }
                    });
}

private void processTextRecognitionResult(Text texts) {
    List<Text.TextBlock> blocks = texts.getTextBlocks();
    if (blocks.size() == 0) {
        showToast("No text found");
        return;
    }
    mGraphicOverlay.clear();
    for (int i = 0; i < blocks.size(); i++) {
        List<Text.Line> lines = blocks.get(i).getLines();
        for (int j = 0; j < lines.size(); j++) {
            = lines.get(j).getElements();
            for (int k = 0; k < elements.size(); k++) {
                Graphic textGraphic = new TextGraphic(mGraphicOverlay, elements.get(k));
                mGraphicOverlay.add(textGraphic);

            }
        }
    }
}

private void runFaceContourDetection() {
    // Replace with code from the codelab to run face contour detection.
}
```

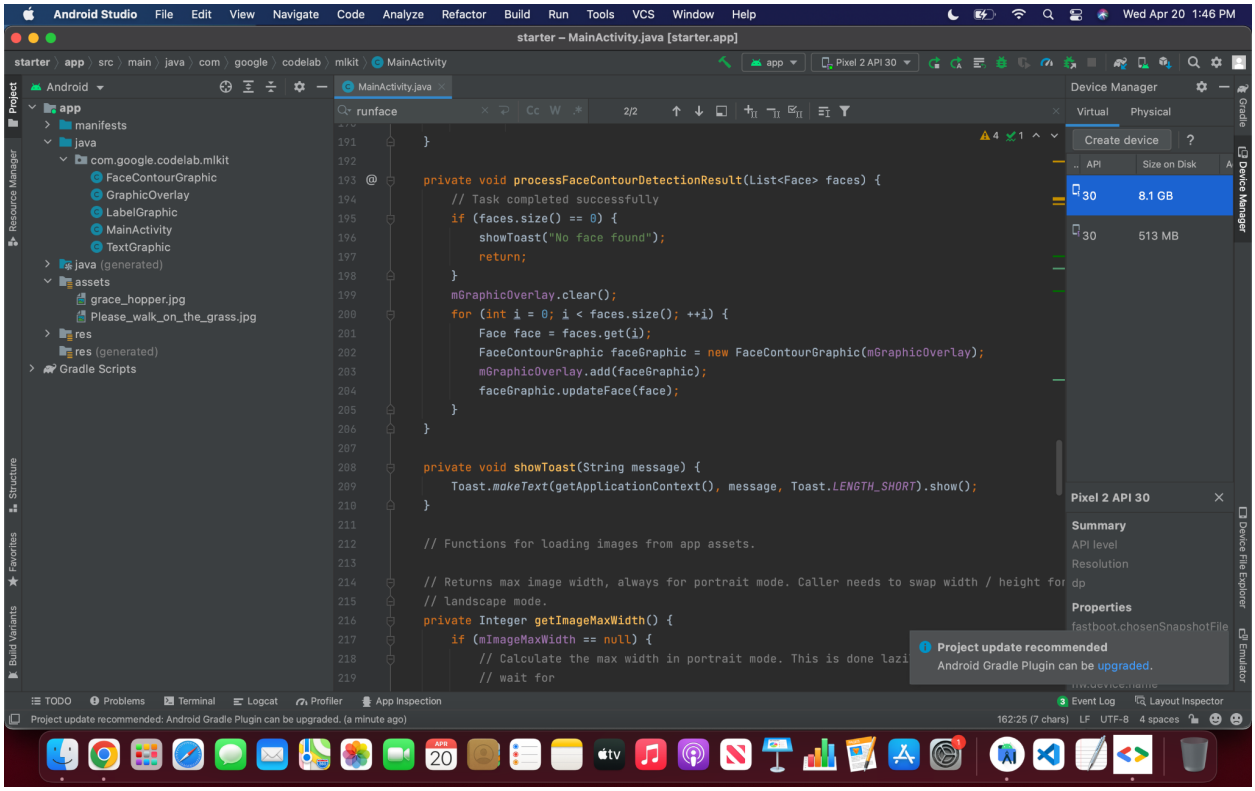
- Run the app on emulator and click find text



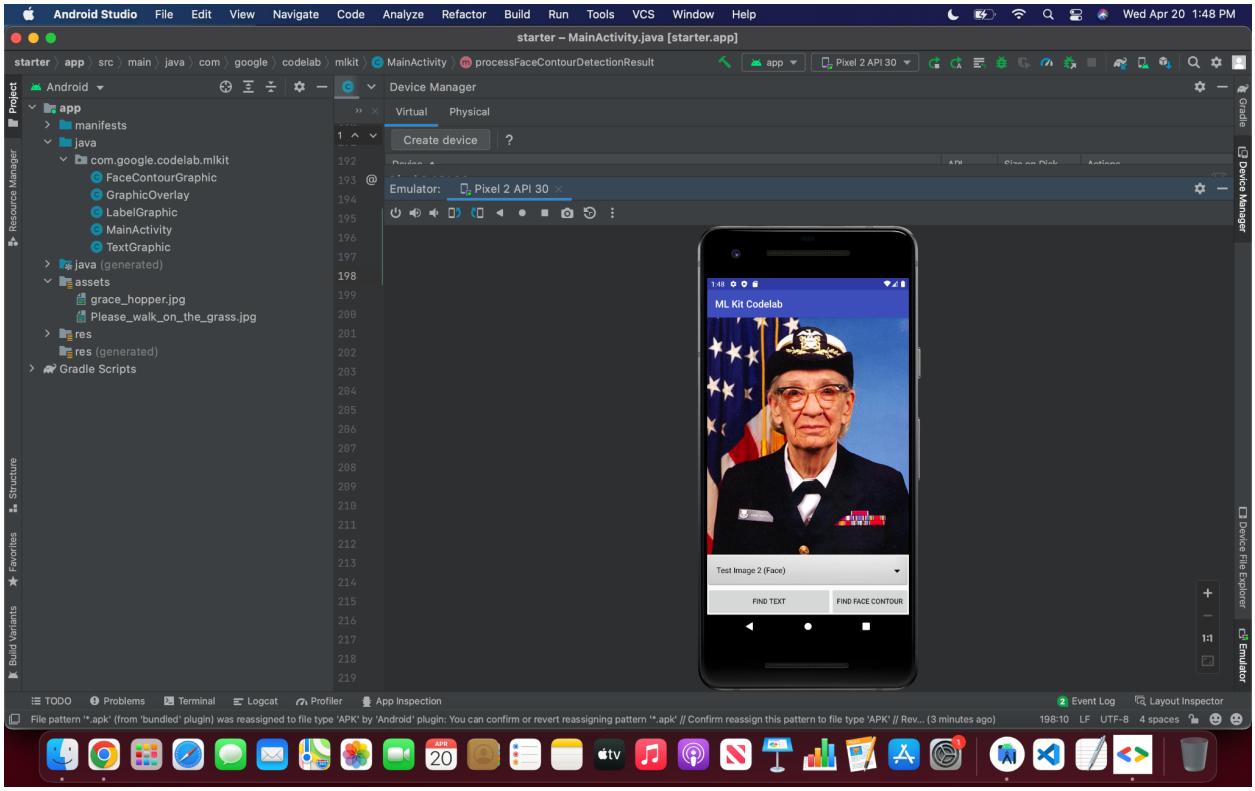
- Add on device face contour detection



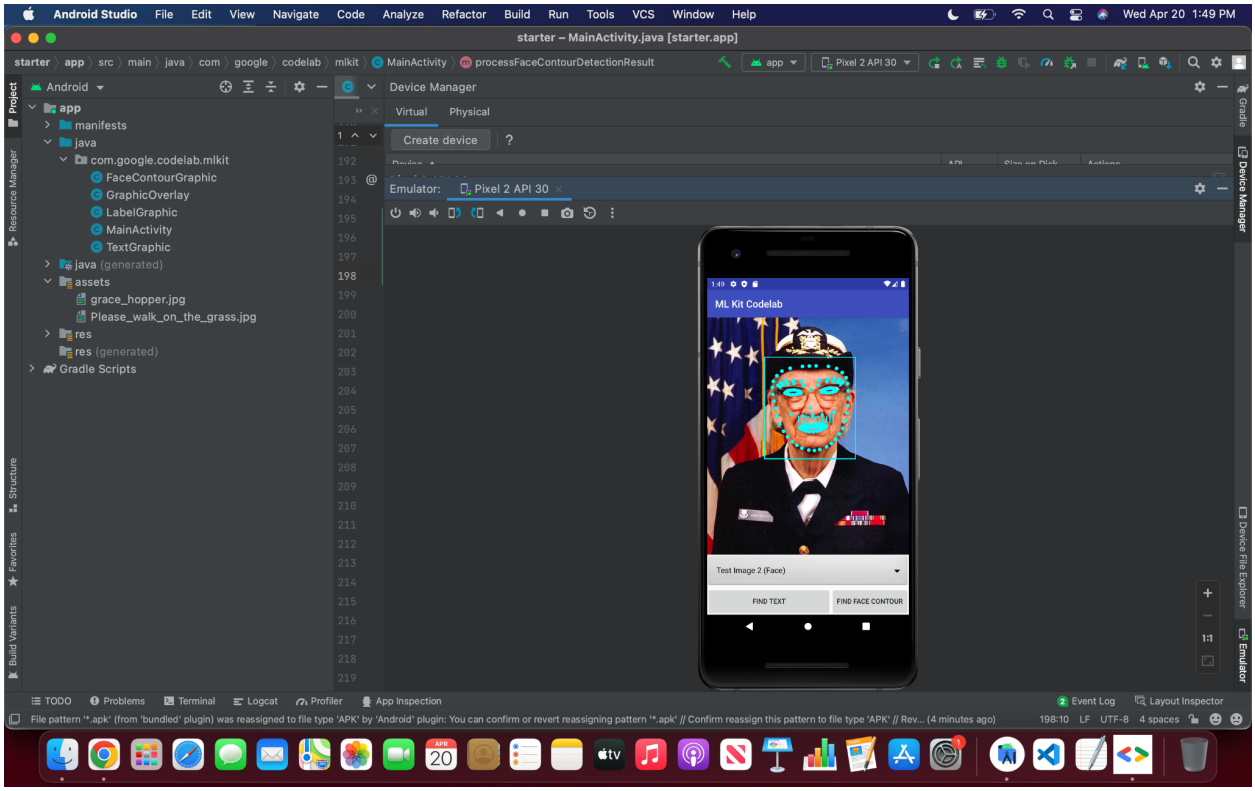
- Update process face contour detection result function



Check the emulator

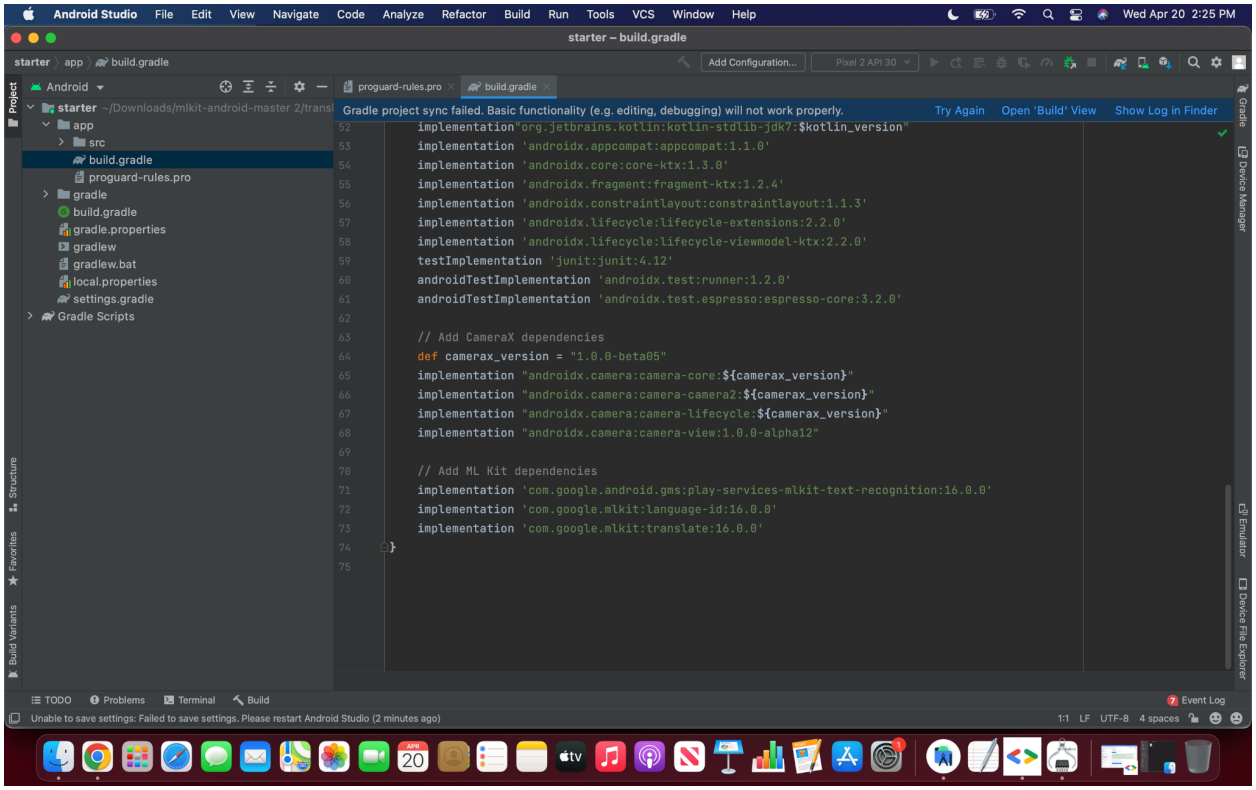


- Find face contour

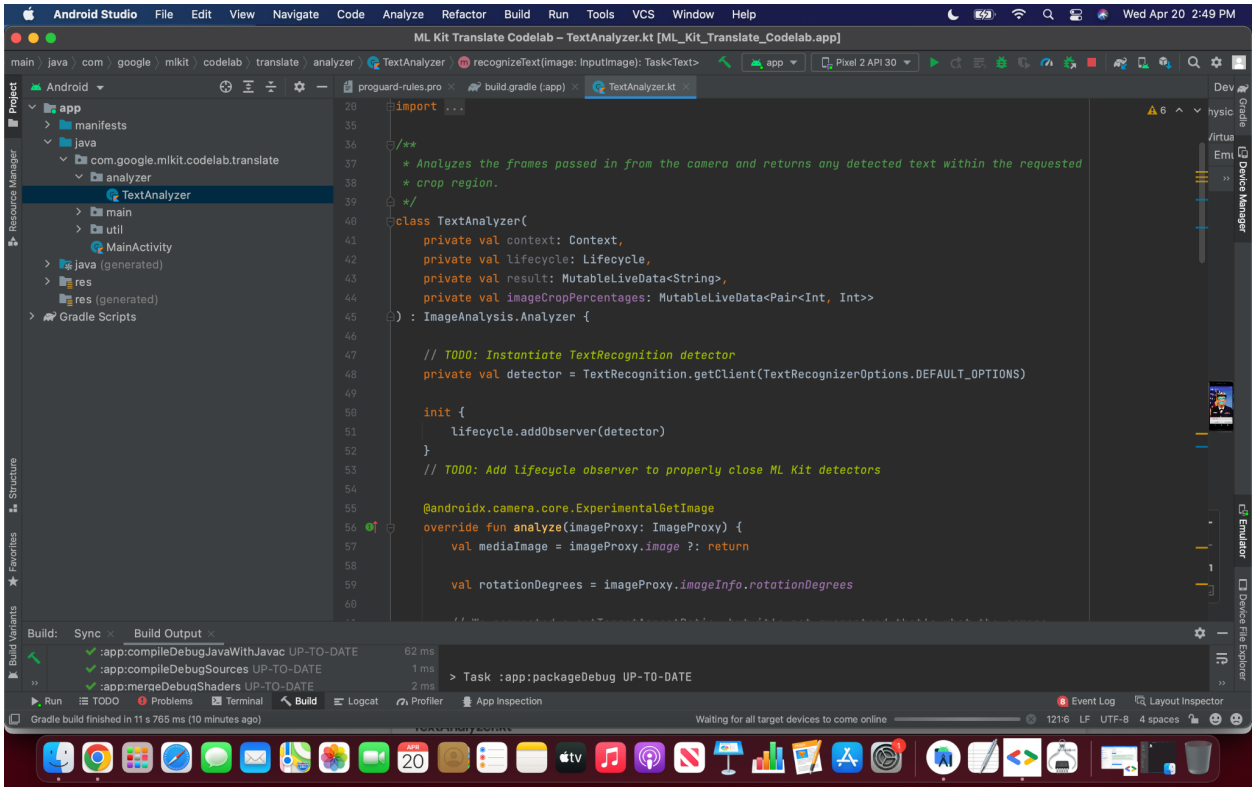


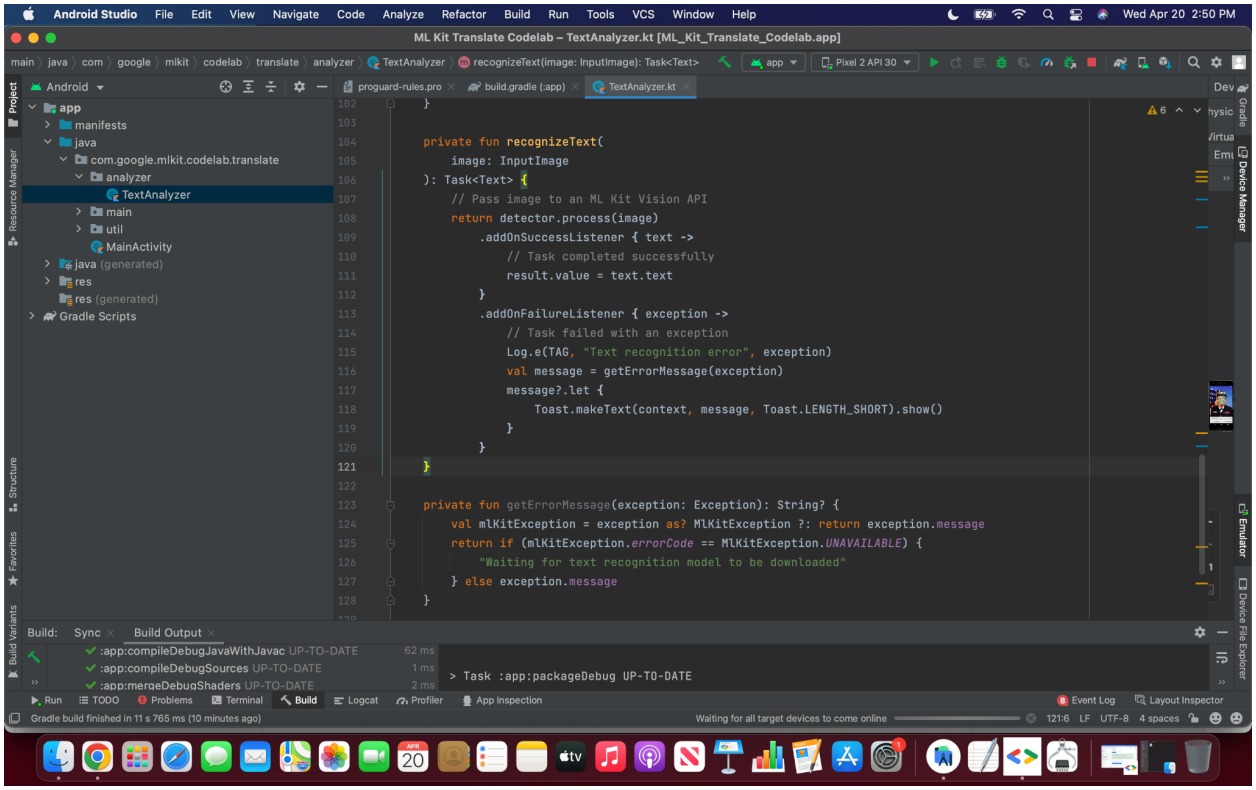
Recognize, Identify Language and Translate text with ML Kit and CameraX:Android

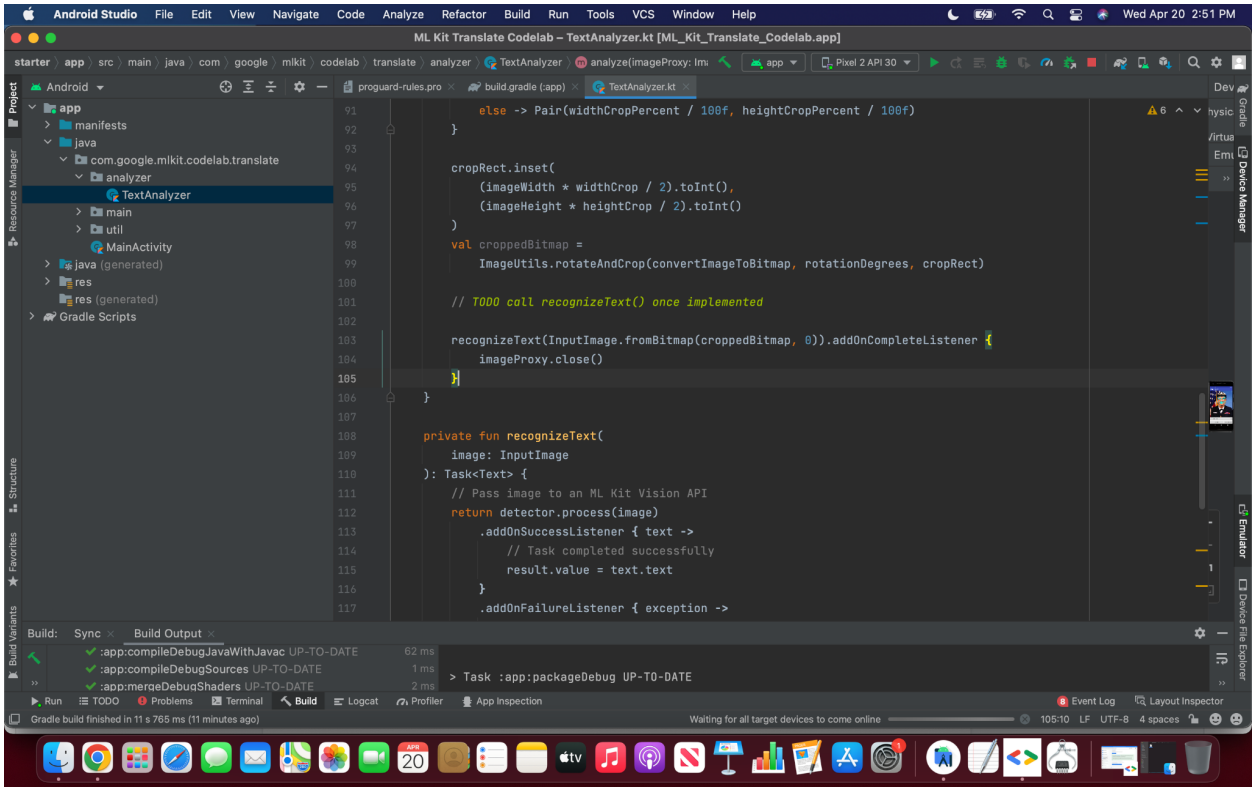
- Import project and check dependencies



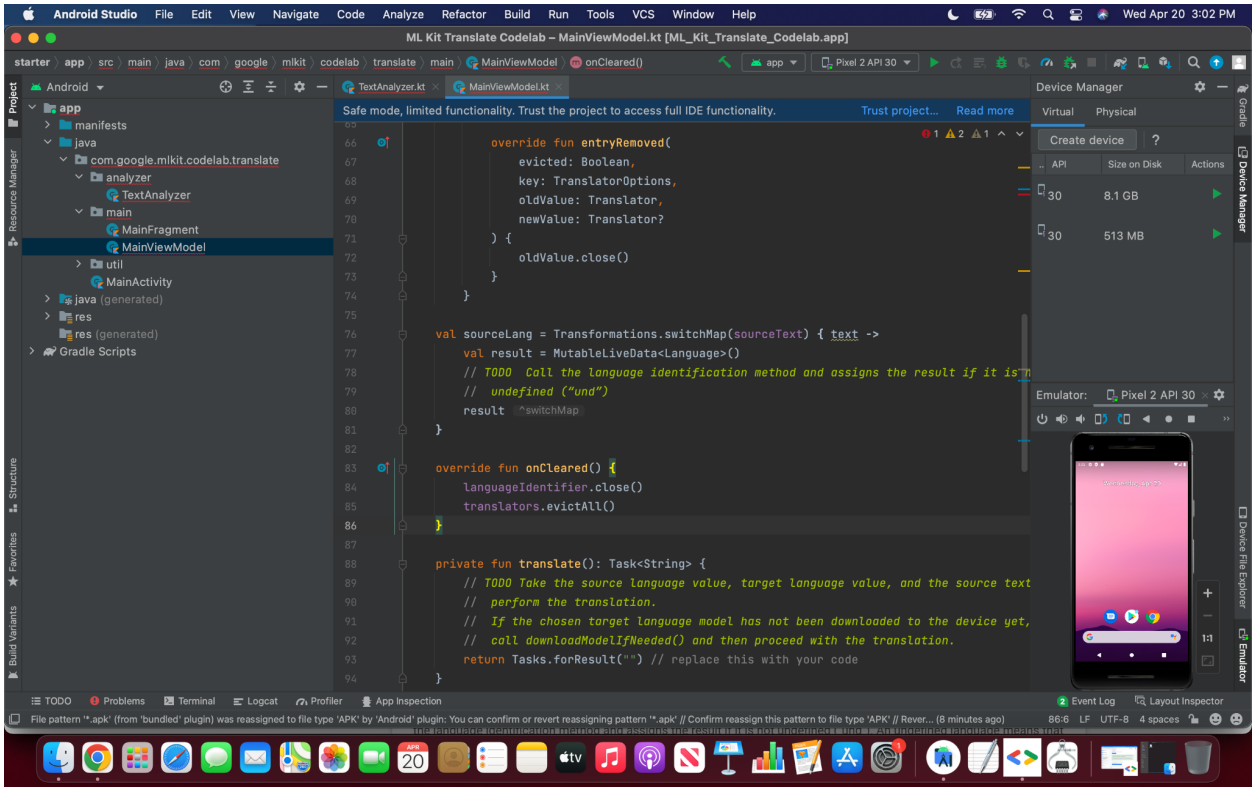
- Update textanalyzer.kt



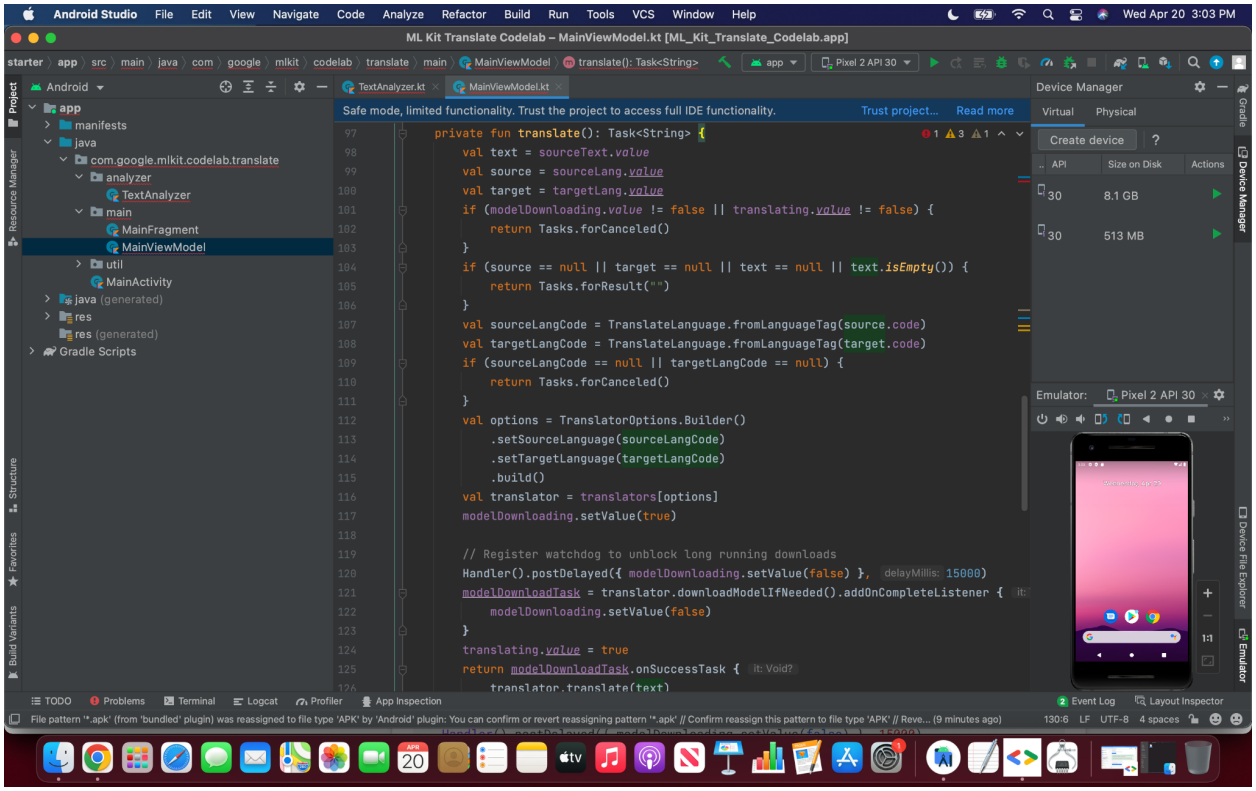

Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Wed Apr 20 2:50 PM
ML Kit Translate Codelab – TextAnalyzer.kt [ML_Kit_Translate_Codelab.app]
main java com google mlkit codelab translate analyzer TextAnalyzer recognizeText(image: InputImage): Task<Text>
app Pixel 2 API 30
Android
app
manifests
java
com.google.mlkit.codelab.translate
analyzer
TextAnalyzer
main
util
MainActivity
java (generated)
res
res (generated)
Gradle Scripts
proguard-rules.pro
build.gradle (:app)
TextAnalyzer.kt
}
private fun recognizeText(
image: InputImage
): Task<Text> {
// Pass image to an ML Kit Vision API
return detector.process(image)
.addOnSuccessListener { text ->
// Task completed successfully
result.value = text.text
}
.addOnFailureListener { exception ->
// Task failed with an exception
Log.e(TAG, "Text recognition error", exception)
val message = getErrorMessage(exception)
message?.let {
Toast.makeText(context, message, Toast.LENGTH_SHORT).show()
}
}
}
private fun getErrorMessage(exception: Exception): String? {
val mlKitException = exception as? MlKitException ?: return exception.message
return if (mlKitException.errorCode == MlKitException.UNAVAILABLE) {
"Waiting for text recognition model to be downloaded"
} else exception.message
}
Build: Sync Build Output
:app:compileDebugJavaWithJavac UP-TO-DATE 62 ms
:app:compileDebugSources UP-TO-DATE 1 ms
:app:mergeDebugShaders UP-TO-DATE 2 ms
> Task :app:packageDebug UP-TO-DATE
Run TODO Problems Terminal Build Logcat Profiler App Inspection
Event Log Layout Inspector
Gradle build finished in 11 s 765 ms (10 minutes ago)
Waiting for all target devices to come online
121:6 LF UTF-8 4 spaces


Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Wed Apr 20 2:51 PM
ML Kit Translate Codelab – TextAnalyzer.kt [ML_Kit_Translate_Codelab.app]
starter app src main java com google mlkit codelab translate analyzer TextAnalyzer analyze(imageProxy: Ima
app Pixel 2 API 30
Android
app
manifests
java
com.google.mlkit.codelab.translate
analyzer
TextAnalyzer
main
util
MainActivity
java (generated)
res
res (generated)
Gradle Scripts
proguard-rules.pro
build.gradle (:app)
TextAnalyzer.kt
else -> Pair(widthCropPercent / 100f, heightCropPercent / 100f)
}
cropRect.inset(
(imageWidth * widthCrop / 2).toInt(),
(imageHeight * heightCrop / 2).toInt()
)
val croppedBitmap =
ImageUtils.rotateAndCrop(convertImageToBitmap, rotationDegrees, cropRect)
// TODO call recognizeText() once implemented
recognizeText(InputImage.fromBitmap(croppedBitmap, 0)).addOnCompleteListener {
imageProxy.close()
}
}
private fun recognizeText(
image: InputImage
): Task<Text> {
// Pass image to an ML Kit Vision API
return detector.process(image)
.addOnSuccessListener { text ->
// Task completed successfully
result.value = text.text
}
.addOnFailureListener { exception ->
Build: Sync Build Output
:app:compileDebugJavaWithJavac UP-TO-DATE 62 ms
:app:compileDebugSources UP-TO-DATE 1 ms
:app:mergeDebugShaders UP-TO-DATE 2 ms
> Task :app:packageDebug UP-TO-DATE
Run TODO Problems Terminal Build Logcat Profiler App Inspection
Event Log Layout Inspector
Gradle build finished in 11 s 765 ms (11 minutes ago)
Waiting for all target devices to come online
105:10 LF UTF-8 4 spaces

- Update MainViewModel.kt



- Update translate method



- Application run