UNIVERSITÉ DES MASCAREIGNES

SAVOIR, C'EST POUVOIR

Université de Limoges

Mental Health Diagnosis

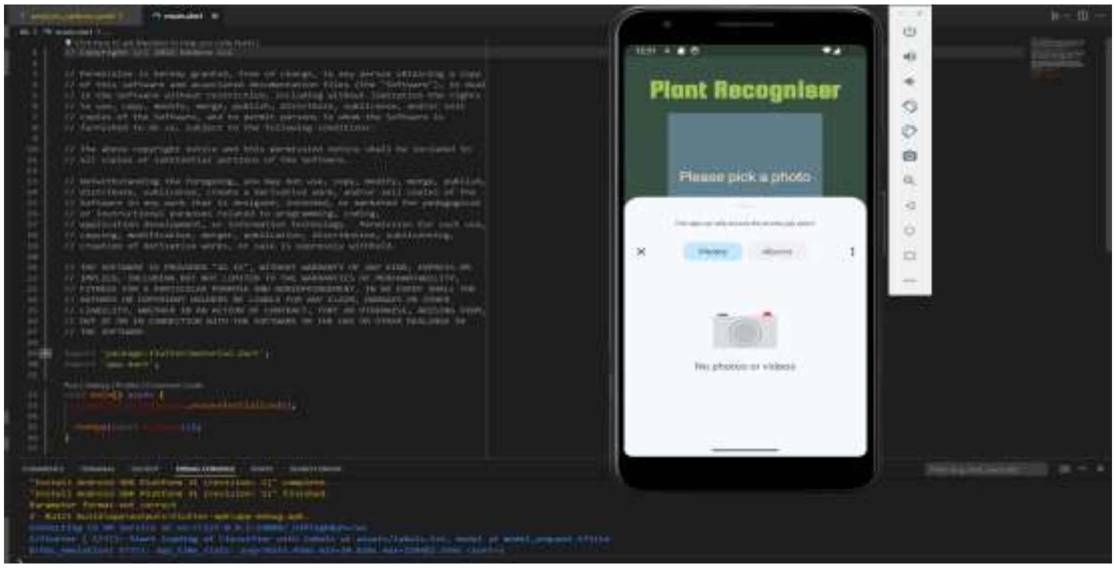
Faculty of Information and Communication Technology

03/02/2024
PMH

Gokhool Diren
Mr Shiam Beehary

TASK 1

Engaging in thorough application testing procedures within the comprehensive and versatile environment provided by Visual Studio Code.



Currently, we are in the process of creating a dataset.

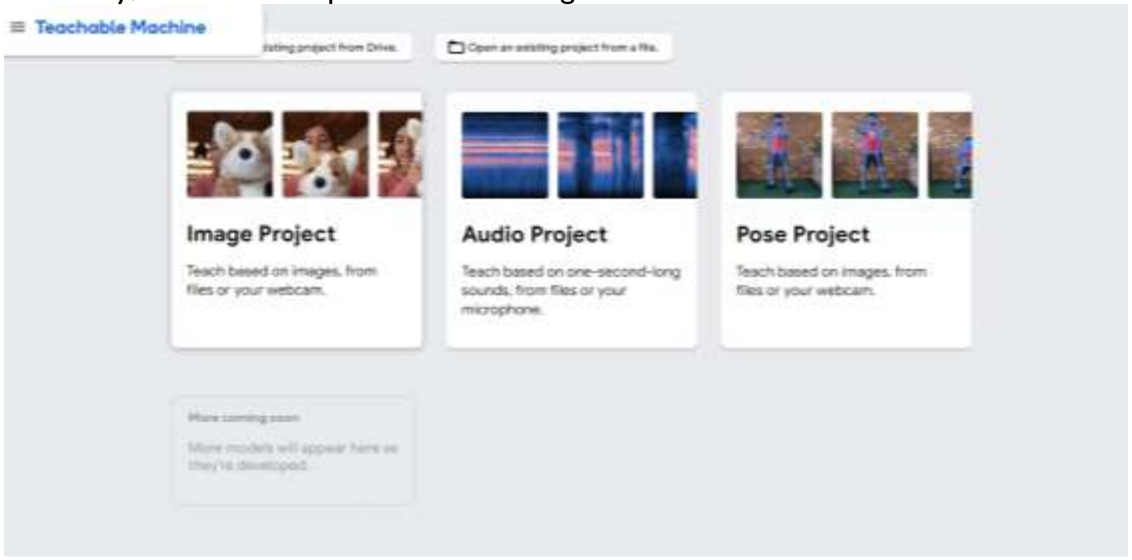


Image Project

Teach based on images, from files or your webcam.

Audio Project

Teach based on one-second-long sounds, from files or your microphone.

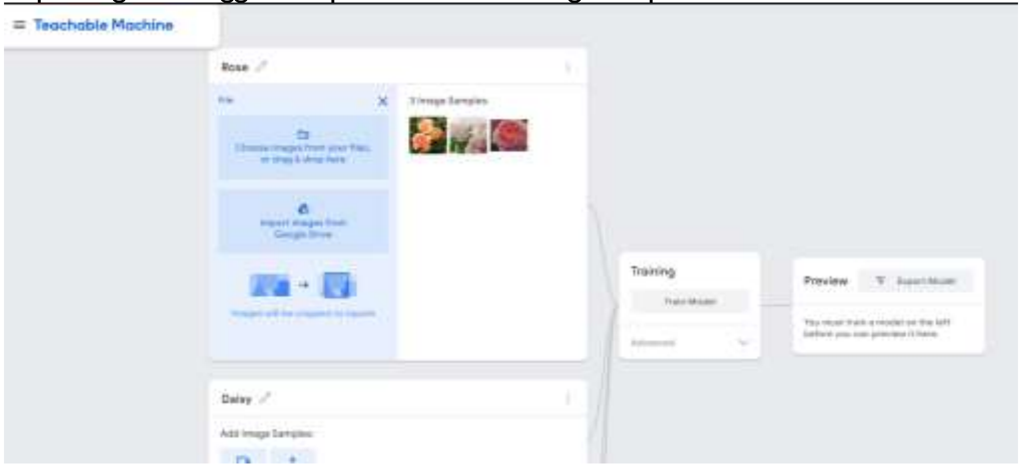
Pose Project

Teach based on images, from files or your webcam.

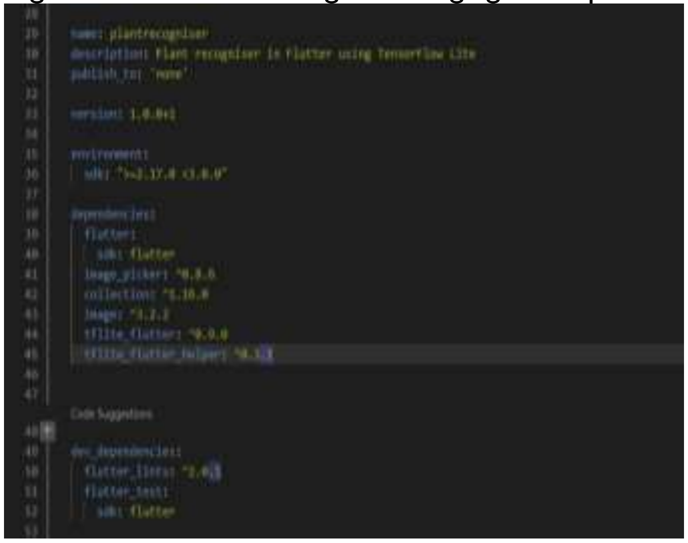
More coming soon

More models will appear here as they're developed.

Exploring the Kaggle sample dataset for insightful patterns and trends.

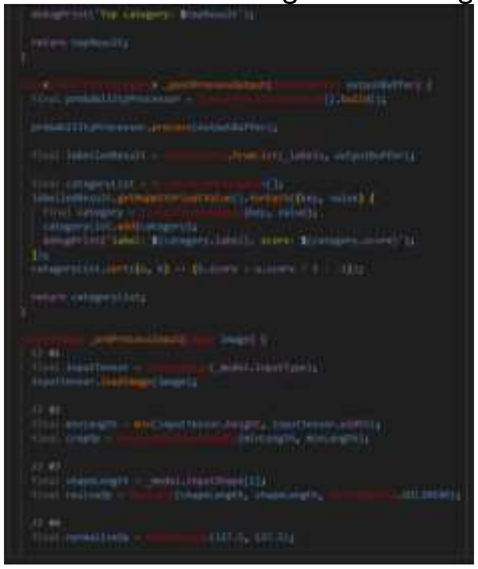


Right now, we're working on changing some parts of the code to make it better.

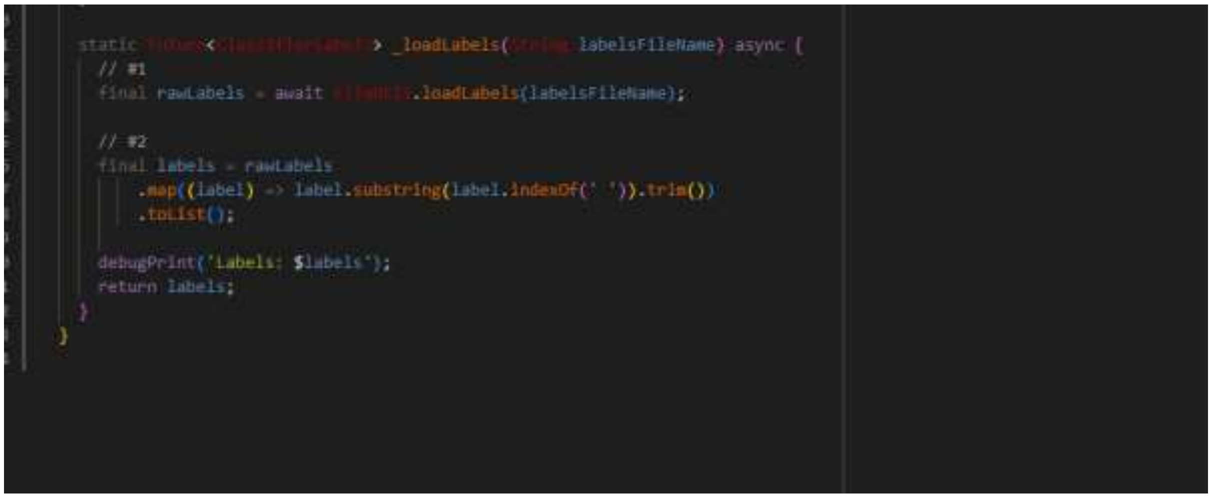


```
name: plantrecogniser
description: Plant recogniser in Flutter using Tensorflow Lite
publish_to: 'none'

version: 1.0.0+1

environment:
  sdk: ">=2.17.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter
  image_picker: ^0.8.5
  collection: ^1.16.0
  image: ^3.2.2
  tflite_flutter: ^0.9.0
  tflite_flutter_helper: ^0.3.1


Code Suggestions

dev_dependencies:
  flutter_lints: ^2.0.1
  flutter_test:
    sdk: flutter
```

We're currently developing code for a classifier, which is a program that sorts data into different categories or classes based on certain features. Additionally, we're working on post-processing techniques for imaging, which involves refining or enhancing images after they've been captured or processed.



```
    debugPrint('Top category: $topResult');

    return topResult;
  }

  List<ClassifierCategory> _postProcessOutput(TensorBuffer outputBuffer) {
    final probabilityProcessor = TensorProcessorBuilder().build();

    probabilityProcessor.process(outputBuffer);

    final labelledResult = TensorLabel.fromList(_labels, outputBuffer);

    final categoryList = <ClassifierCategory>[];
    labelledResult.getMapWithFloatValue().forEach((key, value) {
      final category = ClassifierCategory(key, value);
      categoryList.add(category);
      debugPrint('label: ${category.label}, score: ${category.score}');
    });
    categoryList.sort((a, b) => (b.score > a.score ? 1 : -1));

    return categoryList;
  }

  TensorImage _preProcessInput(Image image) {
    // #1
    final inputTensor = TensorImage(_model.inputType);
    inputTensor.loadImage(image);

    // #2
    final minLength = min(inputTensor.height, inputTensor.width);
    final cropOp = ResizeWithCropOrPadOp(minLength, minLength);

    // #3
    final shapeLength = _model.inputShape[1];
    final resizeOp = ResizeOp(shapeLength, shapeLength, ResizeMethod.BILINEAR);

    // #4
    final normalizeOp = NormalizeOp(127.5, 127.5);
```

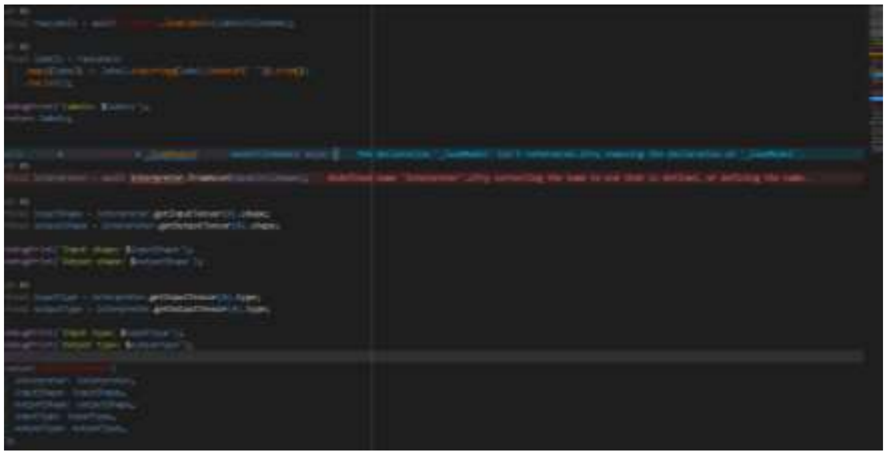


```
  static Future<ClassifierLabels> _loadLabels(String labelsFileName) async {
    // #1
    final rawLabels = await FileUtil.loadLabels(labelsFileName);

    // #2
    final labels = rawLabels
        .map((label) => label.substring(label.indexOf(' ')).trim())
        .toList();

    debugPrint('Labels: $labels');
    return labels;
  }
}
```

At this moment, we're importing a TensorFlow Lite model into our classifier model file. This involves integrating a model that has been optimized for deployment on resource-constrained devices, allowing our classifier to efficiently run on various platforms while conserving memory and processing power.

```
[illegible]
```

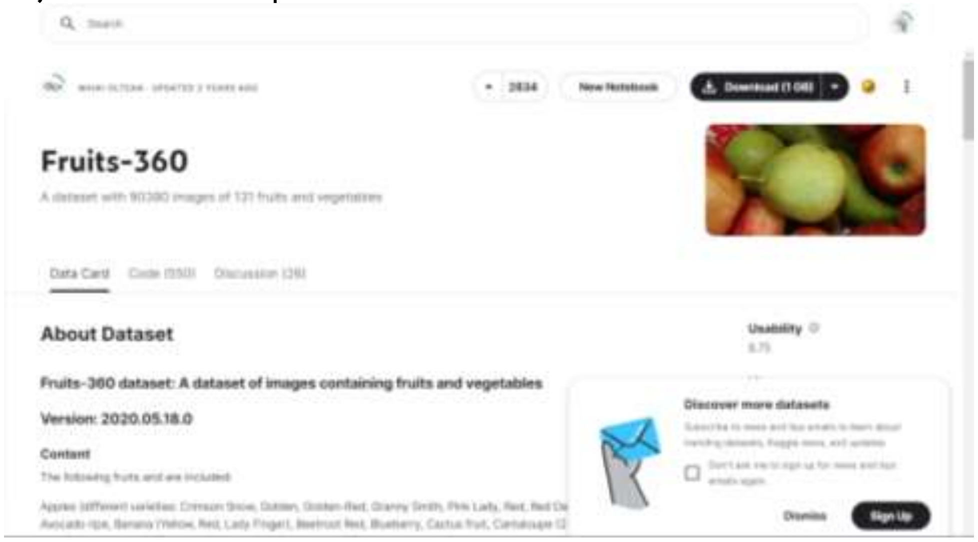
Importing the model from the file



```
[illegible]
```

Task 2

Currently, we're in the process of importing a dataset into Teachable Machine. This involves uploading a collection of labeled data samples, which will be used to train a machine learning model for classification, object detection, or pose estimation tasks.



Fruits-360

A dataset with 90380 images of 131 fruits and vegetables

Data Card Code (350) Discussion (26)

About Dataset

Fruits-360 dataset: A dataset of images containing fruits and vegetables

Version: 2020.05.18.0

Content

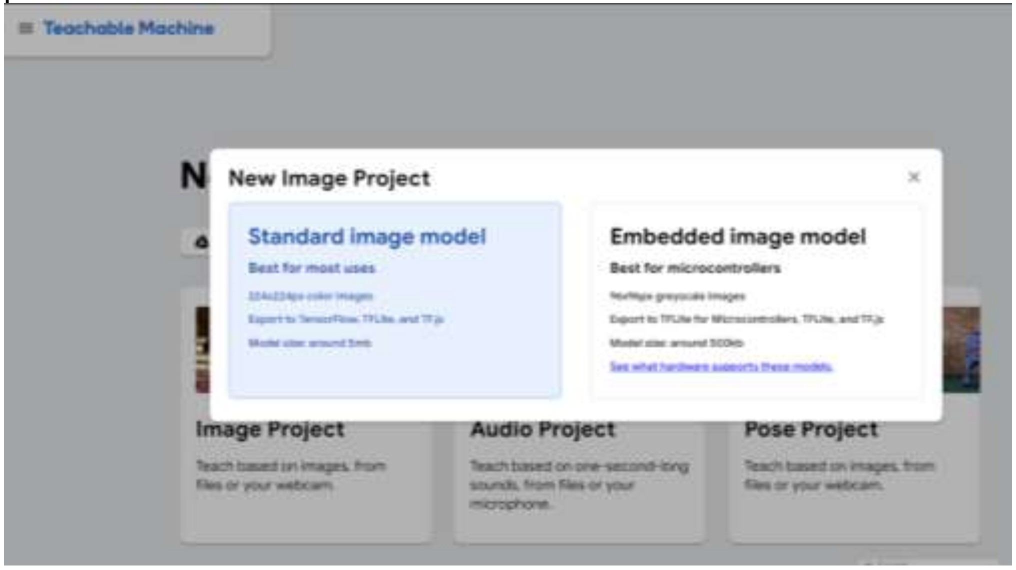
The following fruits and are included:

Usability

8.75

Discover more datasets

Repeating the step of utilizing a standard image model. This involves employing a pre-trained model, such as ResNet or MobileNet, as a foundation for our image classification or object detection tasks. By leveraging these established models, we can benefit from their learned features and optimize our machine learning process.



New Image Project

Standard image model

Best for most uses

224x224px color images

Export to TensorFlow, TFLite, and TFjs

Model size: around 5mb

Embedded image model

Best for microcontrollers

96x96px greyscale images

Export to TFLite for Microcontrollers, TFLite, and TFjs

Model size: around 500kb

See what hardware supports these models.

Image Project

Teach based on images, from files or your webcam.

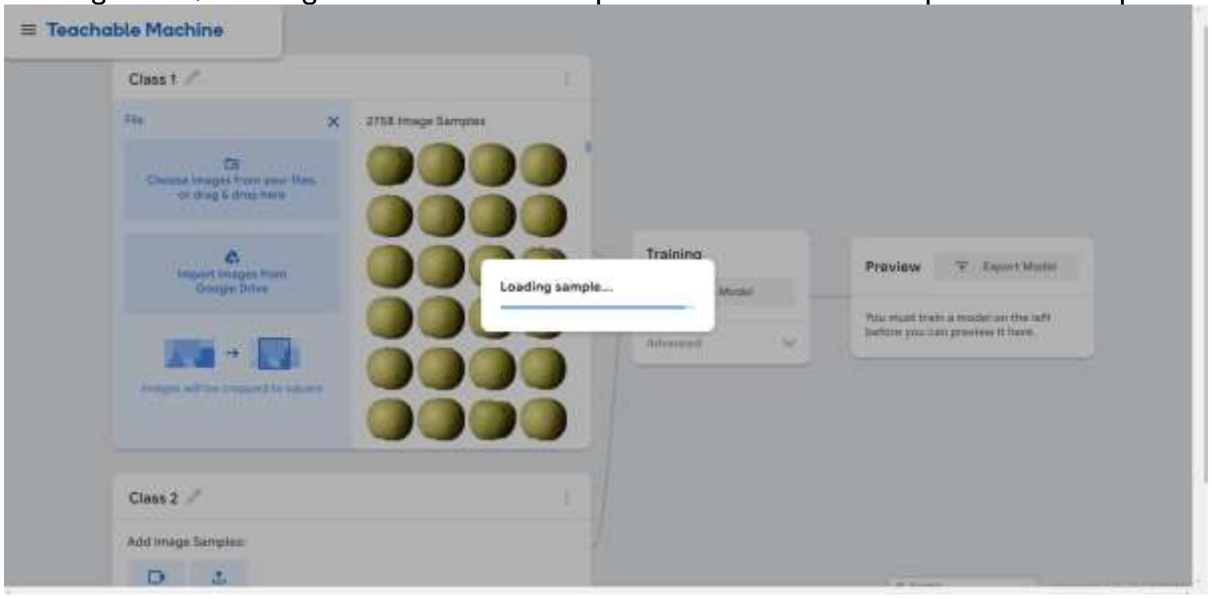
Audio Project

Teach based on one-second-long sounds, from files or your microphone.

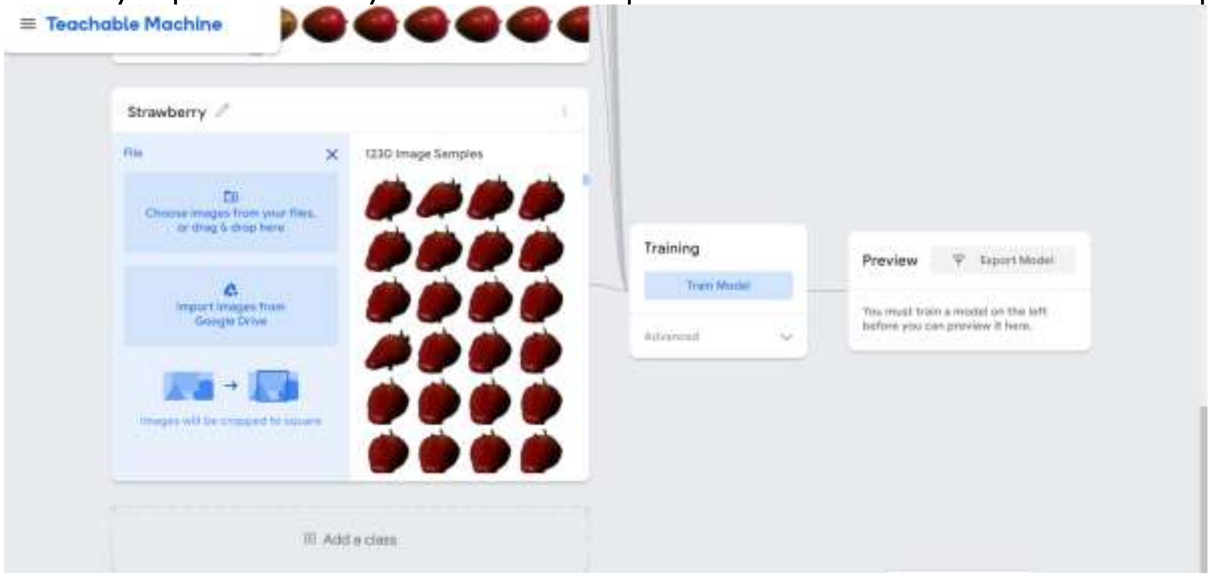
Pose Project

Teach based on images, from files or your webcam.

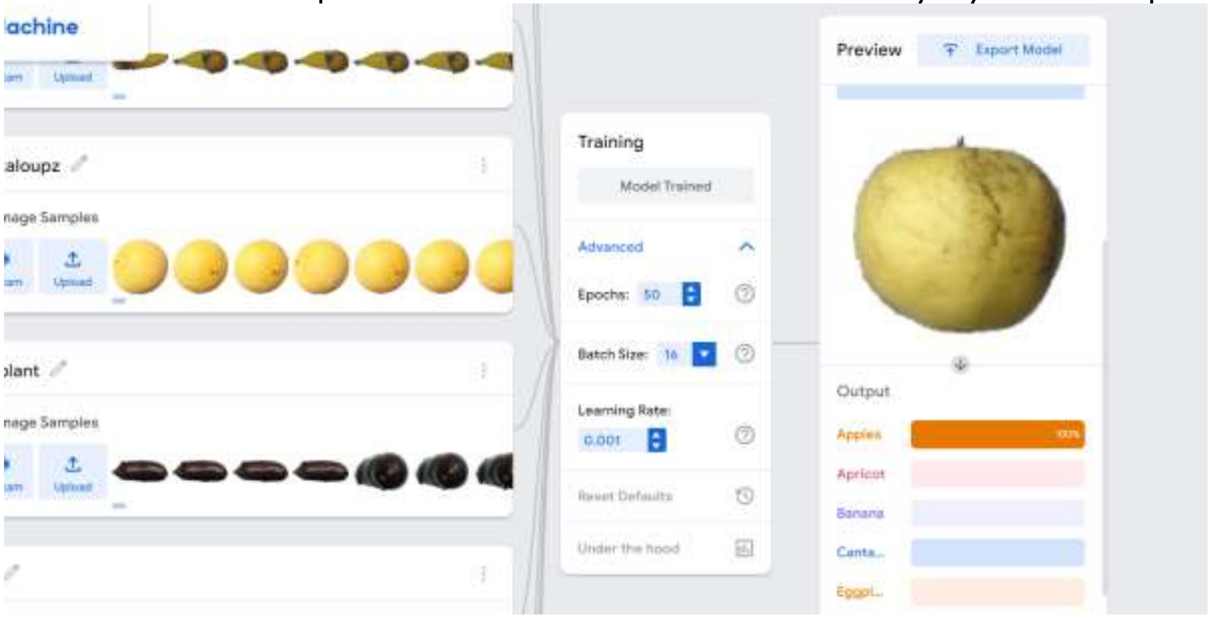
Loading the samples into Teachable Machine. This entails uploading our dataset, which consists of labeled examples, into the Teachable Machine platform. These samples will serve as the training data for our machine learning model, allowing it to learn and make predictions based on the provided examples.


Teachable Machine
Class 1
File
2758 Image Samples
Choose images from your files, or drag & drop here
Import images from Google Drive
Images will be cropped to square
Loading sample...
Training
Advanced
Preview
Export Model
You must train a model on the left before you can preview it here.
Class 2
Add Image Samples:

Training the model. This involves feeding our dataset into the machine learning algorithm, allowing it to learn from the provided examples and adjust its parameters accordingly. Through this training process, the model iteratively improves its ability to make accurate predictions or classifications based on the input data.


Teachable Machine
Strawberry
File
1230 Image Samples
Choose images from your files, or drag & drop here
Import images from Google Drive
Images will be cropped to square
Training
Train Model
Advanced
Preview
Export Model
You must train a model on the left before you can preview it here.
Add a class

Testing the model with samples. This step involves evaluating the performance of our trained machine learning model by providing it with unseen data samples. By examining how accurately the model can classify or predict outcomes for these samples, we can assess its effectiveness and identify any areas for improvement.



Epochs: 50
Batch Size: 16
Learning Rate: 0.001

Apples 100%
Apricot
Banana
Canta...
Eggpl...

Following that, we'll proceed with the same manipulations as those carried out for Task 1. This involves applying similar preprocessing steps, feature engineering techniques, or model optimizations to refine the performance of our machine learning model. By iteratively refining our approach based on the results of our testing, we aim to enhance the overall effectiveness and accuracy of our model.