Machine Learning: Green Line

Problem Story

One shop in Latvia faced a serious issue.

Since this and several other shops has not provided financial support for our olympiad, we'll call it an "undisclosed shop", which may or may not be one of the following: Rimi, Maxima, RV1G kafeinica, Mego, or Depo.

Ivan's friend, who works at this store, met him at the gym and described the problem.

"Some customers mislabel and weigh zucchinis as cucumbers or vice versa. They might do it by mistake (it's realistic to confuse the two) or with malicious intent (because one is cheaper than the other)."

Ivan asked how the current system worked and nearly lost consciousness upon hearing the answer.

"Basically, we take the mean color of what the camera sees."

Being an ML engineer, Ivan spent 10 minutes explaining that's not how ML works.

"Oh wait, did we tell you that we only have 10 images to train on?"

Before the friend could explain himself, Ivan collapsed.

The friend tried to catch him, but his reaction time was even slower than his bench press progress this year.

Luckily, Ivan was unharmed. His friend reported the classification issue to the shop owners the next day. Impressed by Ivan's passionate reaction (and slightly concerned about liability), they called with an offer: substantial funding for Lemon AI if Ivan could fix their model.

Ivan couldn't pass up the opportunity. And neither should you.

Problem

Your task is to train an image recognition model to classify images
into one of the two provided classes: Cucumber and Zucchini.

You are given a remix of a public dataset in the form of several folders with images
corresponding to each of the classes, as well as a baseline model.

The baseline model uses a simple algorithm of averaging pixels on the entire area and performing simple classification on it.

This model, however, needs significant improvement.

Hints

We suggest not to train your own model from scratch. You are suggested to explore materials below to find out a suitable approach for this task.

We know CV may be hard to start with. For that, we prepared a list of materials that may help you get started:

- Very useful video for the task by Hugging Face: https://youtu.be/BqqfQnyjmgg

- 1st and 2nd chapters of Hugging Face's Computer Vision course: https://huggingface.co/learn/computer-vision-course
- The Hugging Face's Image Classification guide: https://huggingface.co/docs/transformers/tasks/image_classification

Scoring

Your solution will be evaluated on a test set remixed from a temporarily undisclosed public dataset. The test set is split into a "public leaderboard" and "private leaderboard".

- The accuracy you achieved on the "public leaderboard" is merely informational.
- The accuracy you achieved on the "private leaderboard" is only visible after the end of the olympiad. It determines your score for the problem.

The score awarded as the task is calculated as follows:

- If your notebook crashes, or the `submission.csv` does not follow the required format, 0 points are awarded.
- Else, let A be the accuracy of your model on the private leaderboard
- let B be the accuracy of the baseline model on the private leaderboard
- score = $max(0, \frac{A-B}{1-B}^{1.5})$

BE AWARE! There are methods of "solving" this problem that are considered cheating:

- Accessing any other data than the train set (including, but not limited to datasets online) for your training.
- Manually labelling the data in the provided test set.

The solutions which break the above rules will be considered invalid, and may lead to disqualification from the olympiad.

Where to Train Your Model?

This task may require computational resources beyond your personal computer.

If that is the case, we suggest using free resources like Google Colab (https://colab.research.google.com/) or Kaggle Notebooks.