Fruit detection for autonomous harvesting systems

1 Introduction

The agricultural sector is increasingly looking towards automation to enhance efficiency, reduce labor costs, and improve yield management. A key area for such innovation is fruit harvesting, a traditionally labor-intensive process. This project is motivated by the potential to develop autonomous robotic systems capable of picking fruit directly from trees and plants. A critical component for such robots is a robust computer vision system that can accurately perceive and locate fruit in complex, natural environments. This proposal outlines a project focused on developing such a system, capable of identifying various fruit types amidst challenging visual conditions like foliage, variable lighting, and occlusions.

2 Goal

The primary goal of this project is to develop and evaluate a computer vision model that can precisely detect and localize multiple types of fruit within images captured in their natural growing environments. The input to the system will be RGB images depicting natural environments with (or without) fruit. The desired output from the model will be a set of predicted bounding boxes for each detected fruit, along with its classified type, aiming to correctly identify every fruit present in an image, as depicted in Figure 1. Successfully achieving this goal means overcoming challenges such as distinguishing fruit from similarly colored leaves, handling variations in fruit size, shape, and ripeness, and coping with partial occlusions by branches or other fruit.

3 Performance measurement

To assess the performance of the developed fruit detection system, it will be primarily used the Mean Intersection over Union (mIoU) metric. This metric provides a comprehensive measure of both localization accuracy and detection correctness. This mIoU will be evaluated in two crucial ways: firstly, an overall mIoU calculated across all fruit instances regardless of type, giving a general

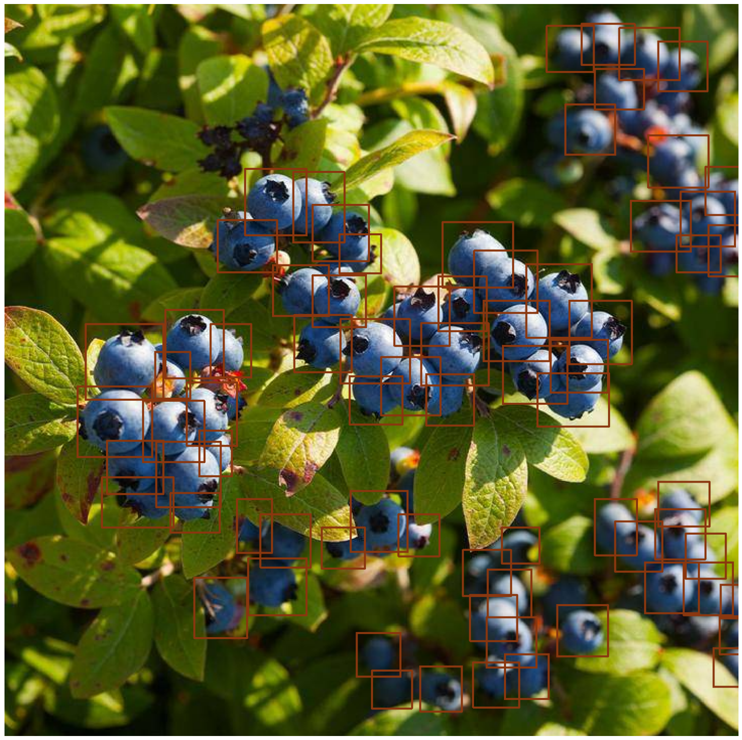

Figure 1: Image of blueberries annotated with bounding boxes.

sense of system performance. Secondly, and perhaps more importantly for practical application, it is useful to calculate per-class mIoU. Also, it's useful to count the total number of fruit for each class appearing in the image. This will help to understand how well the system performs for each specific fruit type, highlighting any particular challenges or successes associated with certain fruit characteristics.

4 Dataset

To validate the developed model, a dedicated test set comprising 27 distinct images is used. These images are carefully selected from a publicly available dataset, specifically the deepNIR Fruit Detection dataset available at `https://datasetninja.com/deep-nir-fruit`. The fruit classes, and the number of images containing that class in the test set provided, are the following:

- cherry (7 images)
- blueberry (4 images)
- orange (4 images)

- kiwi (4 images)
- apple (4 images)
- strawberry (4 images)

Ground truth is formatted with the ”datasetninja” convention, so there are some extra parameters. It is in the form of a json file named as the RGB image, one for each image, structured in the following way:

- description (empty string, not important)
- tags (empty array, not important)
- size (item containing height and width of the image)
- objects (array of fruit in the image)
  - id (serial number, not important)
  - classId (serial number of the class of which belongs the fruit, not important)
  - geometryType (always rectangle)
  - labelerLogin (who labeled the image, not important)
  - createdAt (date of creation of the label, not important)
  - updatedAt (date of update of the label, not important)
  - tags (empty array)
  - classTitle (name of the class, listed above)
  - points (of the bounding box)
    - exterior (array containing the 2 corners of the bounding box, first element is the top-left corner, second element is the bottom-right corner)
    - interior (empty array)