polybee

Computer Vision Task

Given : A set of tomato images captured in a greenhouse are present in the folder along with their annotations and depth maps. The colab notebook provided contains the helper functions to help visualise the annotations and depth maps.

Task 1 : Depth threshold determination

On visualising the annotation using the helper function provided, you may notice that both tomatoes in the foreground and background are detected. Only tomatoes in the foreground must be included for the final tomato count.

Your task is to write a function that automatically determines a depth threshold for an image based on the depth and annotation information provided. (Hardcoded depth thresholds are not acceptable!) There are 7 sets of test cases provided, they have different tomato sizes and are imaged from different distances to the plants.

You may use the below counts as a guideline to check if your algorithm works on all test cases -

- Test1 : 29
- Test2 : 25
- Test3 : ~40
- Test4 : ~48
- Test5 : 14
- Test6 : 13
- Test7 : ~9-11

The script should output the depth threshold, tomato count after applying the threshold and display the image with only foreground tomatoes annotated.

Bonus Task

If you've completed the above task, great job! Here is a challenge for those who want to bag those extra points :

Depth of a tomato could be estimated from a region in the centroid of the tomato -> Obtain centroid from mask, query the points on depth map and determine the average depth.

However, when the tomato is occluded by a leaf, its centroid is erroneously calculated from leaf centroid which leads to errors in depth estimation of the tomato.

Your task is write a function to determine whether a segmented tomato is occluded by a leaf/ branch and estimate its depth from the valid region of the tomato mask (area of the mask not occluded by leaf). The region for estimating depth should be shifted depending on the occlusion.

The script should output the list of tomatoes, their depth and mark the region used for estimating depth on the image.

Your submission will graded based on the following metrics :

- Generalisability of the logic used
- Successful test cases
- Time complexity of the implementation
- Report detailing the thought process, limitations (if any)

General notes :

- Please submit short report of the algorithms used for the tasks, limitations, and if there's any different approaches you'd try to cover all the edge cases.
- A google colab notebook will be shared for the purpose of this task. Please make all your edits in the same notebook for evaluation.
- You can refer to online sources, and you are encouraged to cite the sources used.
- Do send us whatever you attempted within the deadline mentioned in the email.

~~~~~~~~~~~~~~~~~~~~~~~~~~ Happy coding! ~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~