## CSSE463: Image Recognition Project 1: Fruit Finder

**You may work with a partner on this assignment. That is, you may work in “pairs”, as is appropriate for a fruit-finder assignment[[1]](#footnote-1).**

The goal of this assignment is to count the number of objects of different types in an image, where the objects of different types have different characteristic colors. In this case, the objects are apples, oranges, and bananas, which (supposedly) have different colors (red, orange, and yellow). There are three test images on which you are to run your program in this folder: [mixed\_fruit1.tiff](http://www.cs.rochester.edu/~nelson/courses/vision/assignments/mixed_fruit1.gif) , [mixed\_fruit2.tiff](http://www.cs.rochester.edu/~nelson/courses/vision/assignments/mixed_fruit2.gif) , and [mixed\_fruit3.tiff](http://www.cs.rochester.edu/~nelson/courses/vision/assignments/mixed_fruit3.gif) . There is also a very tough image [fruit\_tray.tiff](http://www.cs.rochester.edu/~nelson/courses/vision/assignments/fruit_tray.gif) , which you will use to test the limits of your algorithm.

Your first step is to design a color model. This is just a way to label individual pixels as apple, orange, banana or background. One way to do this is to train a classifier such as a neural network to do this. An easier way, which we will use here, is simply to use thresholds on the color values. For example in HSV space, maybe every pixel within a certain range of yellowish hues that is saturated enough is considered part of a banana.

[Note: For your reference, I have also included pictures of apples, oranges and bananas on a black background taken in daylight and fluorescent light, which theoretically, should assist you in designing color models for the different categories. However, you will find that the colors differ quite a bit based on lighting and will cause your performance on the test images to degrade substantially. Therefore, you will probably end up using the test images themselves to design your final models. In general, that’s not a good principle, but since we aren’t **training** a classifier here to memorize the fruit colors, and it’s your first assignment, I’ll allow it. It also demonstrates the fact that our model is sensitive to lighting, **which you should mention in your writeup**.]

You’ll then have to clean up the image using mathematical morphology operations primarily and any other rules you need to remove spurious background regions. Note that **you may not *manually* fine tune your algorithm to each test image individually**, for example, by hand-coding in that image1 should use 3 erosions and image2 should use 5. (Actually, you could do so if necessary, but since this limits the generalizability of your algorithm, your grade would be “limited” as well.) However, if your algorithm *adaptively* changes thresholds based on something you compute on the fly from the image (like the average size of the fruit), that’s OK; imaging researchers do this all the time. There is a big difference between the two techniques. Ask if this isn’t clear.

You will then use a simple grouping operation to obtain regions with the same label. Connected-components analysis will group adjacent pixels that all have the same value, which will allow you to isolate and count the fruit and find statistics for each.

You can use any procedure in MATLAB to solve the problem, including low-level processing of color images, linear algebra, and statistical operations (if you want them). You are NOT allowed to use a previously-written color segmentation program, or higher level recognition software. This is what you are supposed to do yourself.

**Deliverables:**

1. Your code

2. A detailed writeup, in the format of a conference paper or technical report. Use the sunset paper as an example. **You will be graded according to the rubric in this folder.**

**For example, you need:**

**Abstract:** summarize the paper in 1 paragraph, but include details, numbers, and specific results.

**Introduction**: This is where you convince the reader that he or she should read your paper. Without getting overly technical, you need to address the questions: Why is this a problem? (Convince them it isn’t trivial.) Why do we care about solving this problem? (What is to be gained from a solution?) What is interesting about our work? (If true research, this is where you would differentiate your work from what others have done. If it’s already solved, then at least convince them that your approach is reasonable.)

**Process. Grade is based on completeness of detail.** The first and foremost rule of professional writing is not to skimp on detail. Leaving it until the last minute is no excuse for rushing through a report. Include images along with your descriptions.

Walk the reader through the **details** of your process (but not just reciting the code). For this project, include the thresholds and any other parameters of your algorithm like which morphological operations you used, plus the size and type of your structuring elements. Discuss the difficulties you encountered and how you resolved them. Include **for each** of the test fruit images, the initial test image and each of your intermediate steps:

1. An image showing the initial classification of each pixel using the 3 color models (highlights will be misclassified, some background may be detected, etc., this is OK; you need to show what it looks like before you clean it.). **List your final color thresholds for each piece of fruit. Show detection of all 3 fruit types on one image (partial credit if you show a separate image for each type).**

2. The cleaned image after morphological operations and before any other cleaning operations you did.

**Results**: For each of the test fruit images, show the final image showing the fruit you found (mark these on the image by coloring those pixels that belong to the fruit). Create a table giving the identity, centroids, and size (in pixels) of all the fruit you found.

**Discussion and future work**: Discuss the performance of your algorithm, on these 3 images and the pointing out both successes and failures, and conjecturing why that is the case. Then show how it does on the fruit tray. Discuss any other limitations your algorithm has; I want you to really think about this. Then conclude with a discussion of “Future work”, details of specific things you could do to improve your algorithm if you had more time, both in the short-term (2-3 weeks) and long-term (up to a year).

You will be graded on the quality of your writing, your organization, and the aesthetics of your presentation as well. A good, get- to-the-point resource for writing well is the Handbook for Technical Writers and Editors (<http://www.sti.nasa.gov/publish/sp7084.pdf>). Here are a few of my pet peeves:

Always refer to figures, equations, and tables by number. For example, “Figure 6 shows the foo-ness of the bars. The resulting accuracy of the detector is greater than 90% (Table 2).”

Don’t use contractions. ☺

You are allowed to write in the first person, but write factually, not emotionally. Saying a result is “weird” or that you are happy it works isn’t appropriate.

Know when to use amount vs. *number*, less vs. *fewer*, and much vs. *many*.

Proofread your paper slowly. It helps to have your partner proofread what you wrote and vice-versa. It should take 20 minutes to read carefully and note any errors – anything less and you are just skimming.

**Final hints based on recent observations:**

Don’t shortchange the time devoted to your intro or your discussion. Think about this. Write about it.

Do proofread your paper a final time. I do deduct points for typos and misspellings, and hate to do it when it would be so easy to fix.

Choose your shapes for your structure elements carefully. Is there a better one to use for some of the fruit? Be sure you document your choice in your report.

**Acknowledgement:** this assignment was borrowed in large part from my computer vision professor at Rochester, Dr. Randal Nelson. http://www.cs.rochester.edu/~nelson/courses/vision/

1. I thank John Stephani for this pun. [↑](#footnote-ref-1)