AN APPLICATION FOR FINDING AND CLASSIFYING FRUIT IN IMAGES

Eric Henderson and Nick Kamper

Rose-Hulman Institute of Technology

ABSTRACT

Keywords— image classification,thresholding,median filtering

1. INTRODUCTION

The subject of this paper is an application that will find and classify fruit in a given set of images. The fruits supported are **apples**, **bananas**, and **oranges**.

2. PROCESS

2.1. Overview

The first step in finding the fruit is to convert the image to the HSV image space. Then, simple thresholding is used to remove the background from the image. Next, thresholding is used again, but this time to find individual types of fruit. These initial fruit masks are enhanced by using morphological operators, such as erosion and dilation, as well as a median filter. After obtaining a mask of each fruit, connected objects can be used to find separate pieces of fruit. To filter out any noise that may have gotten past the filtering, the area of each connected region is computed in order to determine any outliers. These outliers are then removed. Finally, the centroid of each piece of fruit and the number of pieces of each type are calculated.

2.2. Thresholding

Our technique involves using two different levels of thresholding: a threshold to remove the background of the image and then a set of thresholds to find the different fruits. This technique allows the foreground filters (for finding the fruits) to be less selective, as those filters will not have to deal with false-positives in the background.

2.3. Enhancing the mask

After thresholding, there is a rough mask of the area where the fruit is. However, these masks contain a significant amount of noise. Our technique uses two methods to reduce the impact of this noise prior to finding regions: a median filter and morphological operations.

First, a median filter, using a 5x5 square, is applied. The median filter is suitable for "salt and pepper" noise, which we determined was one of the more common types of noise seen in thresholded images. Next, a morphological close operation, consisting of an erosion and dilation, is applied to the image using a 3x3 diamond, equivalent to a four-neighborhood. After these operations, the mask is relatively free of major noise and holes.

2.4. Finding connected regions

After the mask is enhanced, our technique will find the connected regions within the image using MATLAB's `bwlabel` using a four-neighborhood. `bwlabel` returns a matrix consisting of 0, 1, ..., n in each cell, corresponding to the regions. These regions are then split up into individual masks and normalized into binary masks.

2.5. Removing more noise

At this point, each mask contains a single connected region. However, there are still some regions that correspond to noise in the mask. To reduce this noise further, our technique will then calculate the average area of each mask and removes those masks which has an area significantly smaller or larger than the average region size. This concludes the processing of the masks by our technique.

3. TRAINING DATA

3.1. Thresholding values

3.1.1. Background capturing thresholds

Filter 1

$$0.000 <= H <= 1.000$$

$$0.275 <= S <= 0.425$$

$$0.100 <= V <= 0.450$$

Filter 2

$$0.150 <= H <= 0.450$$

$$0.000 <= S <= 0.900$$

$$0.000 <= V <= 0.225$$

Filter 3

$$0.100 <= H <= 0.275$$

$$0.125 <= S <= 0.350$$

$$0.000 <= V <= 1.000$$

3.1.2. *Fruit capturing thresholds*

Apple

$$0.925 <= H \textbf{ OR } H <= 0.100$$

$$0.600 <= S <= 1.000$$

$$0.050 <= V <= 0.500$$

Banana

$$0.120 <= H <= 0.250$$

$$0.500 <= S <= 1.000$$

$$0.450 <= V <= 1.000$$

Orange

$$0.060 <= H <= 0.110$$

$$0.600 <= S <= 1.000$$

$$0.500 <= V <= 0.950$$

3.2. Filters and Morphological Operators

1. Median filter using a 5x5 square
2. Close operator using a 3x3 diamond (equivalent to four-neighborhood)

4. RESULTS

4.1. mixed_fruit1.tiff

For the first image, `mixed_fruit1.tiff`, we were able to successfully identify the correct number of fruits.

4.1.1. *Data*

Apple (7 found)

X	Y	Size
143.3259	91.8101	316
48.3462	129.7372	312
208.7509	196.4874	277
186.0201	246.1253	447
218.3883	330.7949	273
60.6172	446.1513	337
65.4917	477.8017	242

Banana (6 found)

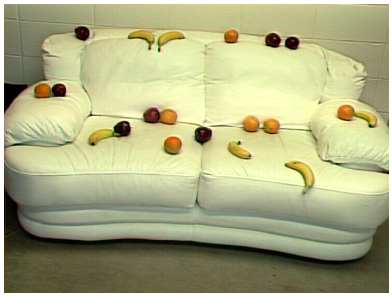
X	Y	Size
218.5945	159.2760	587
52.0207	230.0284	387
55.4043	272.9688	512
242.5063	387.5781	474
279.2613	497.4947	750
187.6559	602.1696	401

Orange (7 found)

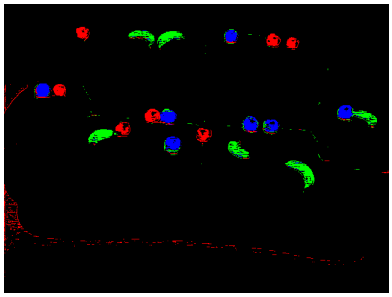
X	Y	Size
143.6724	64.2783	406
187.7799	272.5670	418
231.4619	280.6571	420
53.6546	376.9342	304
200.0660	409.3198	394
202.3764	445.0029	348
179.7903	566.8041	434

4.1.2. *Images*

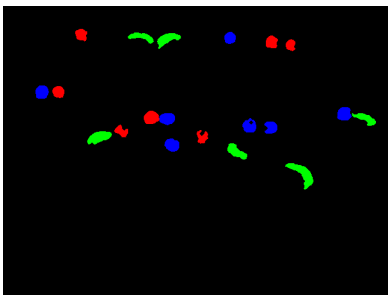
Original image



Original mask



Clean mask



4.2. mixed_fruit2.tiff

For the second image, `mixed_fruit2.tiff`, we were able to successfully identify the correct number of bananas and oranges, but detected an extra apple.

4.2.1. Data

Apple (7 found)

X	Y	Size
369.5518	160.4705	1932
319.0113	266.0188	1864
437.8009	270.6967	211
44.5030	284.1810	1000
407.1295	324.2635	1985
94.5485	484.9597	1537
284.8667	596.4200	300

Banana (6 found)

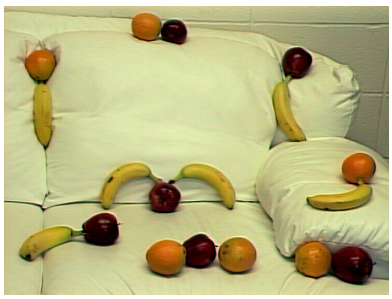
X	Y	Size
388.4214	68.4231	2361
176.4266	64.5710	2494
289.7583	195.3878	2437
290.3219	343.8662	2653
175.9601	467.8580	2557
319.9872	563.1305	2352

Orange (6 found)

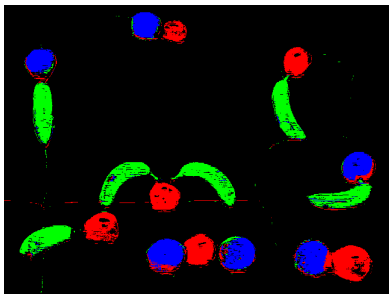
X	Y	Size
92.3523	58.6880	1388
33.0721	234.4446	1498
412.9493	268.4607	2229
414.2446	391.5243	2081
419.1797	511.5228	2215
266.6702	586.3969	1716

4.2.2. Images

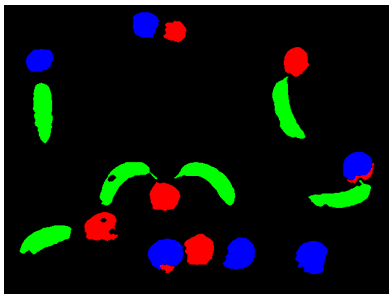
Original image



Original mask



Clean mask



4.3. mixed_fruit3.tiff

For the third image, `mixed_fruit3.tiff`, we were able to successfully identify the correct number of apples, but there was an extra banana and an extra orange.

4.3.1. Data

Apple (8 found)

X	Y	Size
299.3567	36.1389	5657
201.2441	59.2207	639
418.7965	148.1597	5542
287.5883	161.4418	3619
175.9445	184.4902	3874
176.0630	461.1911	2365
298.6333	466.7226	3079
383.6635	539.9577	4583

Banana (7 found)

X	Y	Size
227.2263	99.3108	5476
242.3035	262.0122	4913
76.8940	241.2440	5501
297.6278	293.5087	1150
160.2507	389.4048	5034
61.9554	491.2378	4579
232.1099	575.9033	5004

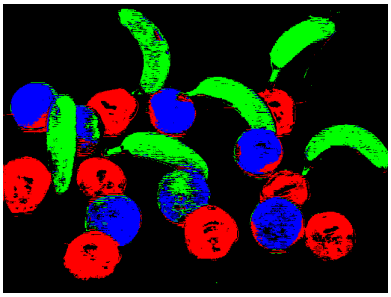
Orange (8 found)

X	Y	Size
165.8371	47.7669	3886
199.9871	133.1239	1162
357.5691	190.3717	4651
179.8290	280.9192	3912
333.2355	277.6628	1210
304.6516	325.1719	1309
242.8631	424.1327	4018
362.9017	456.0160	5067

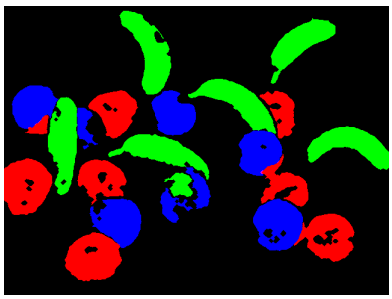
4.3.2. Images

Original image

Original mask



Clean mask



4.4. fruit_tray.tiff

For the fourth image, `fruit_tray.tiff`, we were able to successfully identify the correct number of fruits.

4.4.1. Data

Apple (8 found)

X	Y	Size
310.8944	111.0427	1477
164.9810	138.8308	1897
264.5828	162.8235	2368
373.3612	196.3010	299
124.6049	220.7136	405
203.7126	221.0749	414
265.7575	288.9318	763
228.2927	438.5954	2108

Banana (1 found)

X	Y	Size
82.9100	212.5757	667

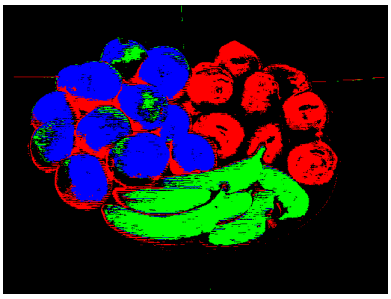
Orange (1 found)

X	Y	Size
229.4551	126.0366	12867

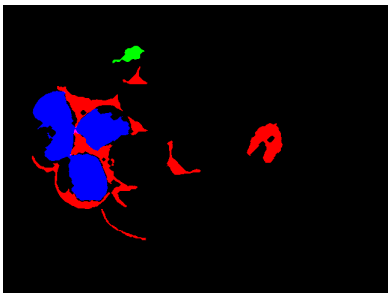
4.4.2. Images

Original image

Original mask



Clean mask



5. FUTURE WORK

There are various techniques that we could use to improve our technique in the future.

5.1. Canny-inspired Adaptive Two-Level Thresholding

One potential idea would be to use multi-level thresholding as a way of capturing more area of the fruits while limiting the amount of additional erroneous data being introduced. The primary idea is to use a "high" threshold to pick out regions that have a high probability of being a fruit, then a "low" threshold would be applied to the neighbors of those "high" threshold pixels and then to other neighboring "low" pixels. This would allow those high probability areas to "grow" out and encapsulate the entire fruit.

5.2. Contrast Improvement

Another potential idea would be to increase the contrast of the image after applying the thresholds to remove the background. This would allow our technique to have a larger difference between neighboring colors, such as the case with the red in the apples, the orange in the oranges, and the yellow in the bananas, which are neighboring colors in the hue band of HSV.

5.3. Shape/Edge Detection

Another potential idea would be to use edge detection to pick out regions that could potentialy be a fruit and use shape detection to determine the type of fruit. This could have better results when combined with color.