

# FRUIT CLASSIFICATION USING MORPHOLOGICAL OPERATIONS ON MASKS

Chirag Sirigere, Connie Zhu

[sirigej@rose-hulman.edu](mailto:sirigej@rose-hulman.edu), [zhuy12@rose-hulman.edu](mailto:zhuy12@rose-hulman.edu)

CSSE 463: Image Recognition

15 June 2021

## ABSTRACT

Recognizing objects in an image can be very useful from securities in public areas to inventory management in store. We developed a system that can count the number of apples, bananas and oranges in different images by applying filters and performing morphological operations on the masks of each fruit. The system will recognize the fruit most of the time, but when the same fruits are stacking together, the system would not be able to correctly count the number of each fruit.

## 1. INTRODUCTION

From e-commerce to traditional industries, companies of all types and sizes can now take advantage of the powerful capabilities of computer vision, which relies on the benefits of innovations in artificial intelligence and machine learning (more specifically, computer vision).

Computer vision has two main applications in inventory management. The first application is that through security cameras, computer vision algorithms can generate very accurate estimates of the remaining items in the store. For store managers, this is very valuable information which can help them immediately detect unusual demand for goods and respond early. Another common application is to analyze shelf space utilization and identify sub-optimal configurations. In addition to discovering wasted spaces, such algorithms can also provide better placement solutions.

In recent years, computer vision is used in fruit harvesting to count the number of fruits in the farm which helps the farmers to better manage the inventory of each fruit.

When harvesting fruit, a video camera is used to capture the moving fruit in the orchard or garden. As

a baseline, we needed to start with images because moving objects are difficult to examine since the camera's angle changes. To begin, three images of fruit with sizes of 480 x 640 pixels were used in classifying the fruit and were converted from RGB to HSV. The difficult part of the task was applying the correct HSV value thresholds to each fruit in the masks such that all three images, or more images, would have the correct count of each fruit with no clustered fruit in the masks. One way to combat this problem was to remove any non-fruit cluster by repeatedly filtering the masks until the number of fruit clusters were unchanged in the masks. The finished, clean masks are then overlaid on the original image.

Eventually, problems rose when one image was used to classify fruits. In Figure 1 below, the difficulty was that the fruits were clustered in a way that it was hard to clearly identify the individual fruit behind the entire cluster of homogenous fruit. At times, the fruit classifier would classify the dark background of the tray as an apple.



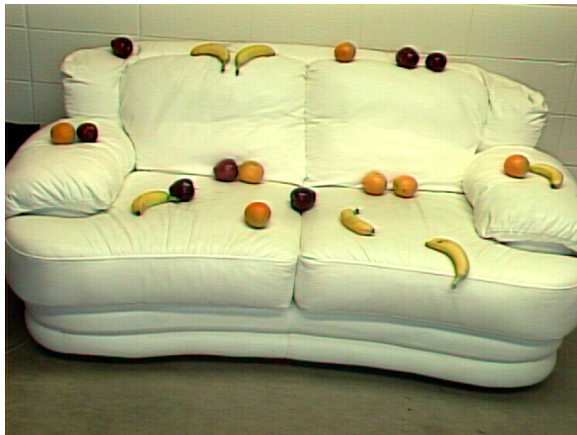
Figure 1: Image of fruit\_tray.tiff [1]

## 2. PROCESS

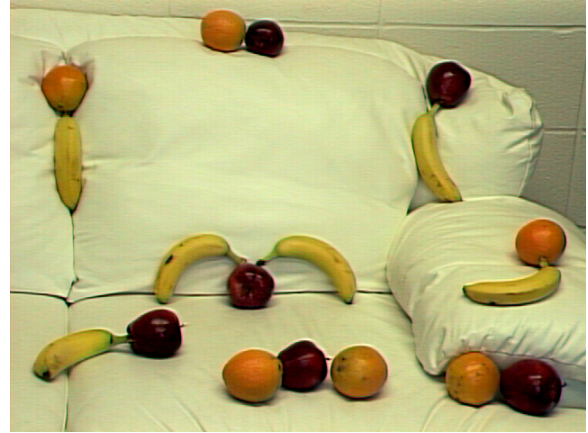
Using images of fruit on a white background couch from the University of Rochester [1], our task was to count and classify the apples, oranges, and bananas in the images. Reading the individual red, green, and blue values for each fruit in any image and classifying the fruits would be a difficult task to handle because the values of the colors are mixed in varying ways such that one fruit is not too red nor too green or blue, but somewhere in between the spectrum. To combat this, a preprocessing procedure was employed and a binary mask was created out of the fruits we were trying to find in the image. A post-processing filter and morphological procedures were used to clean the image. After that, the individual fruit's masks were displayed on the original image.

### 2.1 Data Source

The folder of fruit images are all 480 x 640 pixels encoded in the TIF file format with three slices of color: red, green, and blue. The fruit photos are from Randal Nelson, Department of Computer Science in the University of Rochester.



**Figure 2:** Image of mixed\_fruit1.tiff [1]



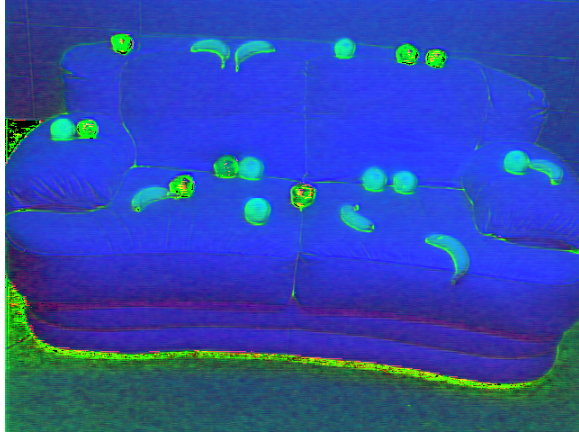
**Figure 3:** Image of mixed\_fruit2.tiff [1]



**Figure 4:** Image of mixed\_fruit3.tiff [1]

### 2.2 Preprocessing

The 480 x 640 RGB color values were converted to 480 x 640 HSV values and no resizing was done. An example of the RGB-to-HSV converted image taken from Figure 1 is shown.



**Figure 5:** Image of hsv\_mixed\_fruit1.png, the HSV converted image of mixed\_fruit2.tiff

### 2.3 Fruit pixel models

To create a mask for the individual type of fruit in the image, the background of the image was set to zero and the HSV pixel values of the fruits that were found were set to one, serially, based on the pixel threshold models provided in Table 1 and Table 2.

The pixel models were found by manually iterating through the HSV thresholds until they reached the

desirable values in Table 1 and Table 2. The number of pixels that were classified as noise was determined manually by looking at the binary image mask.

Another way we decided on the threshold values was by zooming in on one fruit and displaying the HSV value histogram to determine the pixel count ranges for each fruit. This gave us a definitive value range for a fruit, but only for that fruit, and the value ranges could not be applied to other images with different colored backgrounds and fruit.

**Table 1:** HSV Thresholds for each fruit from Chirag Sirigere.

Fruit	Hue	Saturation	Value
apple	[0, 0.1)	(0.5, 1]	[0, 0.4)
orange	[0, 0.12)	(0.6, 1]	(0.6, 1]
banana	(0.12, 0.2]	(0.6, 1]	(0.6, 1]

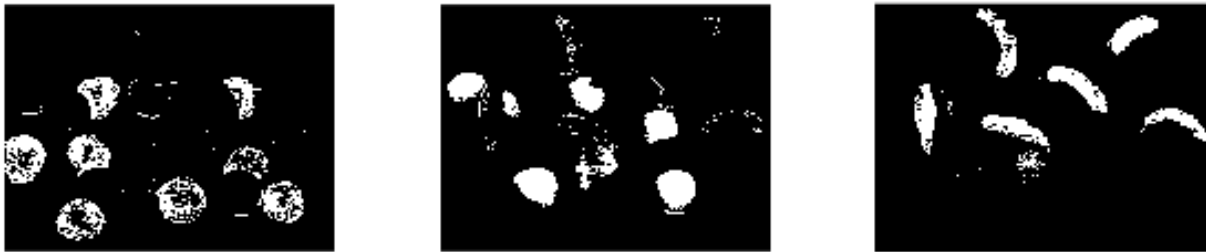
Applying the HSV values from Table 1 to the mask for each image gave masks with small clusters of pixels that would not be classified as fruit, for example the couch as an apple in the bottom and left portions of Figure 6.



**Figure 6:** Image of masks1.png with correctly classified fruits and noise. Shown from left to right: apples, oranges, bananas from mixed\_fruit1.tiff.



**Figure 7:** Image of masks2.png with correctly classified fruits and noise. Shown from left to right: apples, oranges, bananas from mixed\_fruit2.tiff.



**Figure 8:** Image of masks2.png with correctly classified fruits and noise. Shown from left to right: apples, oranges, bananas from mixed\_fruit3.tiff.

In Figures 6, 7, and 8, we noticed that other fruits were falsely taken into account for the masks as well, especially their edges. In Table 2, similar HSV thresholds were found, which gave similar results in the masks for each image. The varying HSV values later help us determine the filter kernel values and the size comparison factors.

**Table 2:** HSV Thresholds for each fruit from Connie Zhu.

Fruit	Hue	Saturation	Value
apple	[0, 0.1)	(0.5, 1]	[0, 0.4)
orange	[0, 0.12)	(0.7, 1]	(0.4, 1]
banana	(0.13, 0.2]	(0.6, 1]	(0.4, 1]

## 2.4 Morphology and post-processing

Applying masks on images shows the position of the fruits in the image. However, there is also too much initial noise that needs to be eliminated.

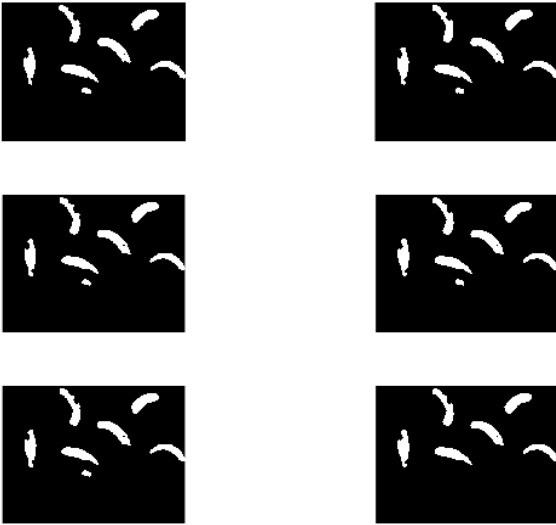
One way to eliminate this noise is to apply median filters to the mask. The kernel for the filter we used was  $9 \times 9$  which eliminated most of the noise while keeping the correct regions of the fruits.

After applying a median filter on the mask, we clustered each region to find the average size of all the regions and eliminated the clusters that are less than a fraction of the average size of the clusters. This factor varies and depends on each fruit, but we chose a factor of 8 because any lower would eliminate a correctly identified fruit and any higher would leave in noise that would need further elimination.

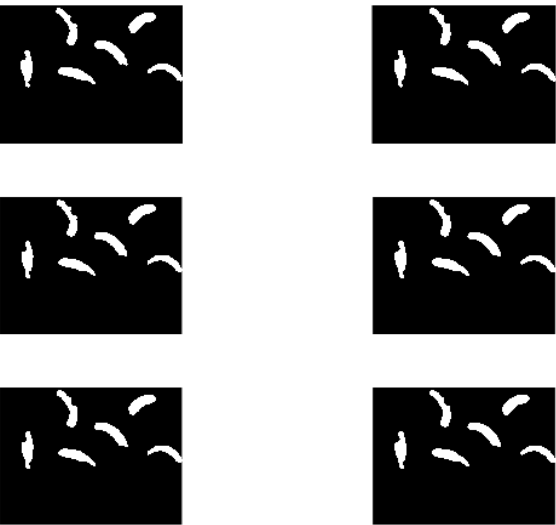
The masks were then fed through morphological operations such as closing, dilating, and opening, using a  $3 \times 3$  arbitrary structural element before being fed through a size comparison again.

The entire process is done in a while loop where the terminating condition is a mask that has no change in the number of clusters of the same fruit that it detects after each filter, size comparison, close, dilate, open,

and another size comparison. This is shown in Figure 9 and Figure 10.



**Figure 9:** Image of the first iteration of morphological operations on the bananas from Figure 8



**Figure 10:** Image of the second iteration of morphological operations on the bananas from Figure 8

### 3. RESULTS

The resulting images in Figures 11, 12, and 13 show the regions of the fruits which were found by the program



**Figure 11:** Shown from left to right: the correctly found apples in red, oranges in orange, and bananas in yellow from Figure 2, mixed\_fruit1.png



**Figure 12:** Shown from left to right: the correctly found apples in red, oranges in orange, and bananas in yellow from Figure 3, mixed\_fruit2.png



**Figure 13:** Shown from left to right: the correctly found apples in red, oranges in orange, and bananas in yellow from Figure 4, mixed\_fruit3.png

**4. DISCUSSION**

The system finds fruits and counts a small number of individual fruits correctly on the images where the fruits are placed separately and not together. The system is able to filter out all the noise, leaving the region of the fruit on the mask.

The main problem with the system is that the edge detection of fruits is not accurate enough when the same fruit is placed together. Since we are separating fruits based on their HSV values, the system would recognize a bunch of apples as one apple because the

color of the apple is too similar to its neighbor which also happens to the bunch of bananas and oranges.

The second problem with our system is that even though it recognizes the fruit and counts the number of the fruit correctly, it only recognizes the fruit based on counting the number of regions after the mask is filtered and dilated, rather than recognizing them based on the shape of the fruit. If the regions are linked together, it recognizes it as a single fruit. If the region disappeared, the system would not be able to know that it lost a fruit. This happened because we

applied the same filter to different images, but the size of the fruit is different in each image. Therefore, one filter will not be enough for different images.

## **5. CONCLUSIONS AND FUTURE WORK**

Since the detection of the edge of each fruit is not accurate enough, in the future, we can use edge detectors such as the Canny Edge detector in our system and perform morphological operations on the edge of the fruit to get a more accurate edge. The solution could potentially solve the problem of missing a fruit or recognizing the fruit as a different fruit. In addition, the solution could also solve the problem where the same fruits are placed together. Using the Canny Edge detector, we would thin the edges where there is a shadow of the tray, fill and round the masks such that they enclose the entire fruit and trim any excess noise using erosions and median filters. If we had another year to work on this problem, we would use machine learning algorithms such as k-means and convolutional neural networks on the images.

## **REFERENCES**

[1] Images courtesy of Randal Nelson, Department of Computer Science, University of Rochester, accessed March 2006.