Digital Image Fundamentals

Vilmorin Project

Cristian Rocha
Yat Hong Lam

January 2017

General Work Flow

1. Database of melon images with scale and color chart
2. Fixing the orientation of melons
3. Segmentation
    a. Segment the melon area from the background
    b. Removal of the stem of melon
    c. Segment the 1cm coin
4. Labeling
    a. Determine the geometry of melons
    b. Determination of maximum rectangle in melon
5. Determination of bold fringe on melon and selection of interior part
    a. Segmentation of bold fringe with regionprops
    b. Get the convex hull region of major regions
    c. Build masks to extract convex hull regions
    d. Classification
6. Segmentation of netting from the melon background
    a. K-means clustering on HSV color space
    b. HSV Thresholding
    c. Otsu thresholding
7. Feature extraction
    a. GLCM
    b. Harris Detector
    c. HOG
8. Machine Learning and Classification of melons

Files

Orientation correction - **orientation2.m**
Determination of max rectangle in melon - **maxRect.m**
Batch processing to correct orientation and crop max rectangle - **batchProcess.m**
Melon segmentation, remove of the stem determining melon`s geometry - **melons.m**
Coin segmentation - **segmentCoin.m**
Get the size of the major and minor axis of the melon - **getSize.m**
Batch processing to get the center window - **getCenterWindow.m**
Batch processing to get the window of 3 major region - **getWindow3.m**
Function to get the center major region of a image - **imgProcessingSem.m**
Function to get 3 major regions in a image - **imgProcessingSem3.m**
Crop the convex hull of a region - **maskConvexCropped.m**
Find the center region - **centerWindow.m**
Machine learning and feature extraction - **training.m and testing.m**

Fixing the orientation of melons

By observing the current configuration of image, the orientation of the image is mixed, shown in FIg.X. To facilitate the further analysis, it is better to fix the orientation. The first orientation is considered as the preferred orientation. In such situation, the white cardboard of melon name and QR code is the brightest part in the image, which is placed at the lower left corner of the image. The correction process is based on this fact.

1. The whole image is divided into four quadrant region.
2. The image is converted into gray level image.
3. In each region, histogram with 4 bins is created.
4. The ratio of pixel in the last bin (the highest intensity) in the image are compared in the four regions, the region with highest region is considered as the one which contains the white cardboard.
5. Use imrotate to rotate the image to wanted orientation.

Fig.1 Different orientations of image. A rotation algorithm is used to correct them to the preferred orientation, which is the first one.

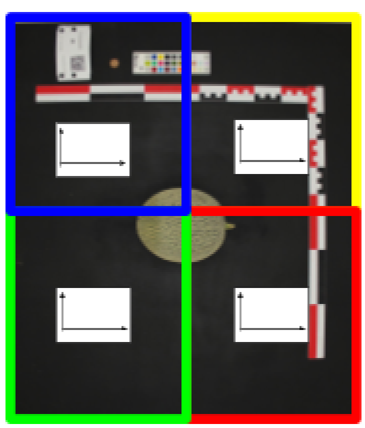

Fig. 2 Schematic of rotation correction step. In the example, the blue quarter shows the highest proportion of bright pixel and should be rotated to the lower left corner.

Segmentation

For the segmentation part of this project it was required to segment the melon from the background area, remove the stem from the melon and segment the coin in the image. The procedure developed in order to accomplish these tasks is described below.

Melon Segmentation

The first thing what can be observed in the database of images is that all the melons are *greenish* or *yellowish*. On the other hand, the coins, the ruler and the colour chart have different colours - with exception to some parts in the colour chart. Taking this into account, one trivial way for segmenting only the melon is by thresholding in the HSV colour space. For the H channel, the threshold is set for values only in the green and yellow area. In the saturation channel, we do not get values with low saturation - less than 10% of saturation. Similarly, for the V channel, the threshold is for values above 10, which means we do not get values that are too dark.

In order to improve the accuracy of the segmentation, we apply the closing function and we can get a more consistent melon. The result can be seen in the image below.

Figure 3: Segmented melon

The threshold is not perfect, as there are other objects in the image which also have similar colours to the melon. However, after the thresholding, the melon will always be the biggest element in the image. With that in mind, we use region props from MatLab for getting the biggest element in the image. Then, we apply a mask with the original image in order to obtain the segmented melon with its colour.

This method works well for most of the images. However, it is not accurate for the melon in figure 4. This problem is probably due to that the threshold does not get all the pixels in the left border of the melon - as the colour in this area is not exactly consistent with the colour we set in our threshold. Another problem for the segmentation of this melon is that its form is really different from other melons, as a result, the closing function, using a disk as structure element, will not perform well due to the difference of the structure element and the form of the melon.

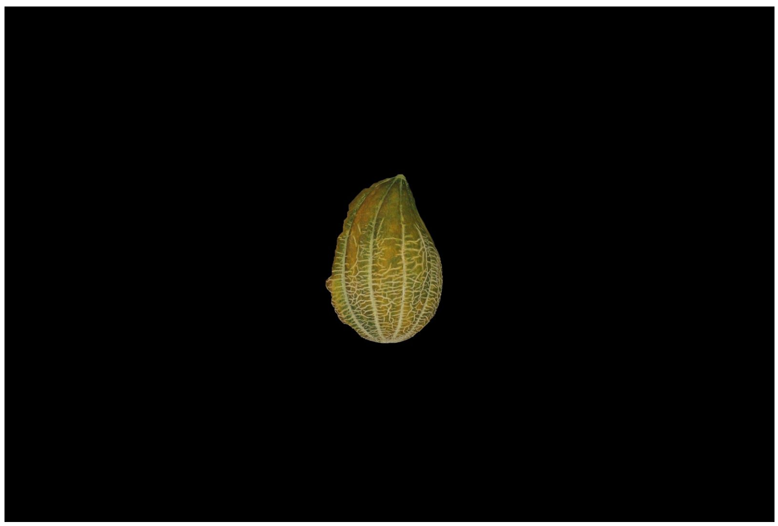

Figure 4: Not accurate segmented melon

Removing of the stem of the melon

The procedure for removing the stem is quite simple. After having the segmented melon, we apply the opening function with a disk as structure element. With that, the stem is always separated from the melon, and by region props, we apply a mask to select only the biggest element - which is the melon. The result of this procedure is in figure 5.

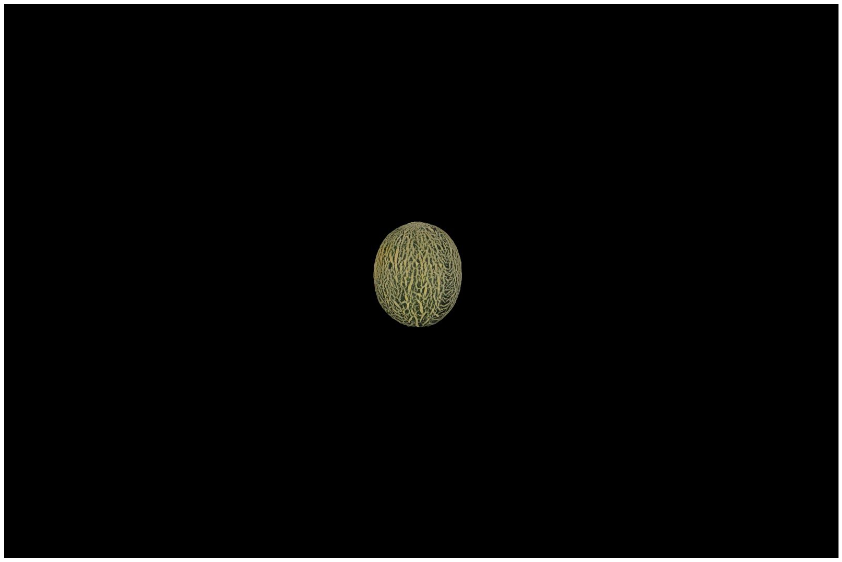

Figure 5: Segmented Melon without stem

Segment the 1cm coin

This method is also straightforward if you take in consideration that the coin will always be the element with the smallest eccentricity of the image. A circle for example has eccentricity zero, and the eccentricity increases when the object is less “circular”. As the coin is the most circular element of the image, we use region props to label all the elements and calculate the eccentricity of the objects. Based on the eccentricity of all the objects, we select the one which is the closest to a circle - which will always be the coin.

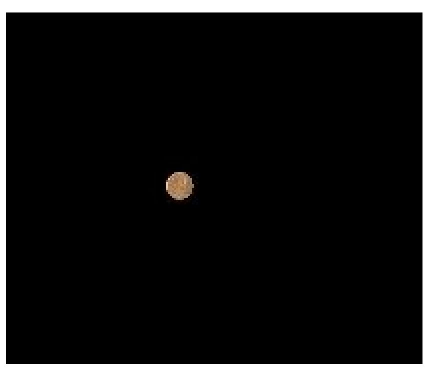

Figure 6: Segmented coin

Labeling

Determine the geometry of melons

In order to get the geometry of the melon, we developed a function for extracting the major and the minor axis of the melon by region props. As we know what is the diameter in cm of the 1 cent coin, the diameter in pixels of the 1 cent coin and the major and minor axes in pixels of the melons we can obtain the major and the minor axis of the melons in cm by a simple rule of three:

$$\frac{diameter\ in\ pixels\ of\ coin}{major\ axis\ in\ pixels\ of\ melon} = \frac{diameter\ in\ cm\ of\ coin}{major\ axis\ in\ cm\ of\ melon}$$

$$\frac{diameter\ in\ pixels\ of\ coin}{minor\ axis\ in\ pixels\ of\ melon} = \frac{diameter\ in\ cm\ of\ coin}{minor\ axis\ in\ cm\ of\ melon}$$

Table 1: formulas for obtaining the major and minor axis in cm

Figure 7: Melon classified as elliptical by our method

With the major and the minor axes it is easier to determine what is the geometry of the melon. If the major and the minor axes are similar, it means the melon is more circular; otherwise, it means the melon tends to an elliptical form - we are assuming the melon can be only elliptical or circular. The other forms are not take in consideration.

Figure 8: Melon classified as circular by our method

Determination of maximum rectangle in melon

The black background affects the feature extraction process. It is better to focus on the interior area of the melon without any background.

As melon is either spherical or ellipsoidal, they look like circle or elliptical in 2D. Just as the previous part, the major axis and minor axis of melon can be determined by regionprops. Then we can make use of differentiation to obtain the largest rectangle. From the mathematical derivation, the size of rectangle can be found as:

$$x = \frac{a}{\sqrt{2}},\ y = \frac{b}{\sqrt{2}}$$

Then the rectangle is put in the centroid of melon, which can be determined by regionprops.

In most of the situation, it can give a fairly good result. But sometimes it can fail as the melon is not a perfect ellipse in 2D. Sometimes black background can be found in the rectangle. A further cropping can be used to improve the result.

1. Count the number of black pixel, N,in the black-and-white image of the cropped area.
2. Black pixels are assumed to distributed in one square (see the red square) with length $\sqrt{N}$, and it can be appeared in either one of the four corners.
3. The new rectangular area (shown in yellow) having width $x - 2\sqrt{N}$ and height $y - 2\sqrt{2}$ put in centroid is cropped.

This algorithm can effectively get a rectangular region with melon only.

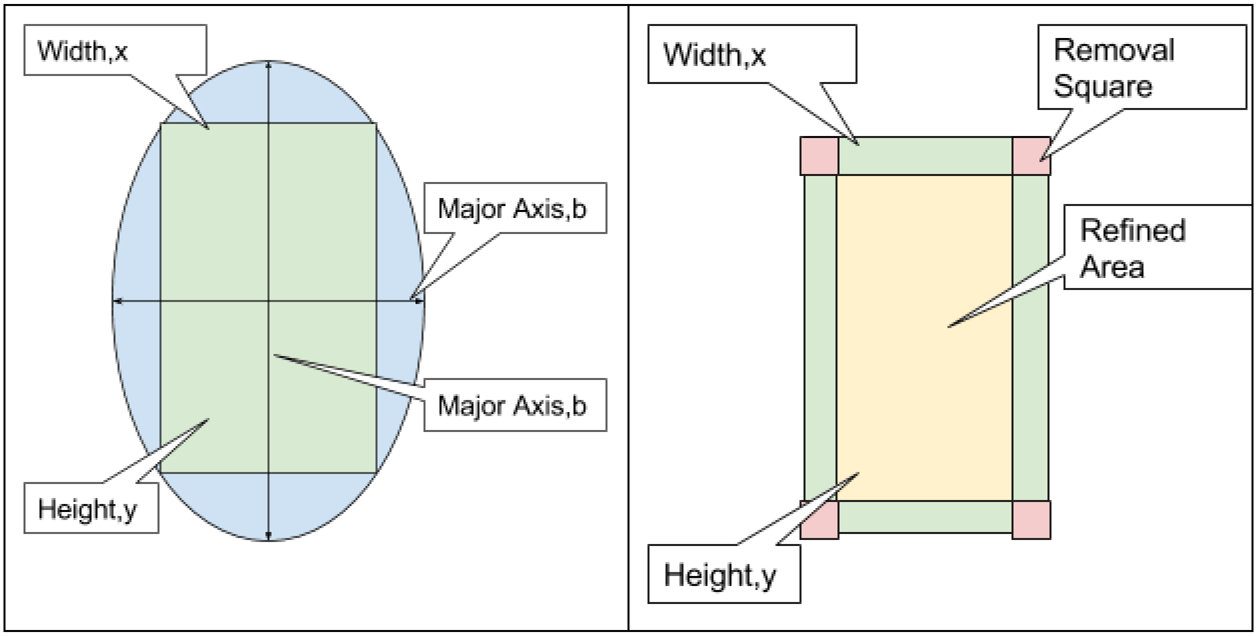

Fig. 9 Schematic diagram to obtain the maximum rectangle.

Determination of bold fringe on melon and selection of interior part

From our observation, in the maximum rectangle of melon, there are mainly three components:

1. Bold fringes
2. Nettings
3. Background

The existence of bold fringes and nettings on the melon surface is optional. According to our observation, the netting density is not related to the area with fringes so the bold fringe area should be ignored. However, the color in fringe often affect the further segmentation step so it is better to remove them and only select a window that contains background and nettings.

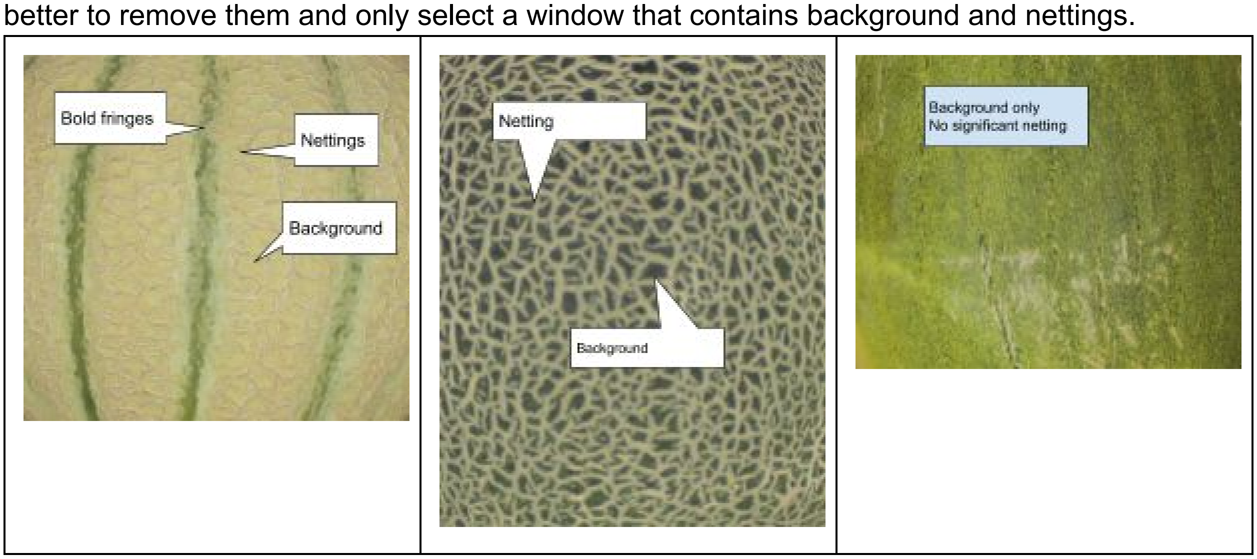

Fig.10 Bold fringes, netting and background.

If fringes exist, then the background and nettings are separated in regions by bold fringes. These regions can be extracted by following algorithm.

1. By histogram equalization and grayscale thresholding, a black and white image is obtained.
2. Use imfill to fill small holes in image
3. Use regionprops to get the largest three region in the figure.
4. As the region can be in an awkward shape, the convex hull region is selected.

The result is shown in Fig.11. It can successfully obtain the netting and background regions if fringes exist. Under such situation, the largest three regions have similar area ration in original image. Otherwise, the algorithm give a single major region and all minor regions are very small. The area proportion can be a useful indication to verdict the existence of fringe.

The irregularity of region can be a problem for further and affect the fairness of judgment. By some testing, we decide to crop a window of 100x200 from the region center for later analysis. The schematic is shown in Fig.12. Alternatively, sometimes only the window located nearest to the center is wanted.

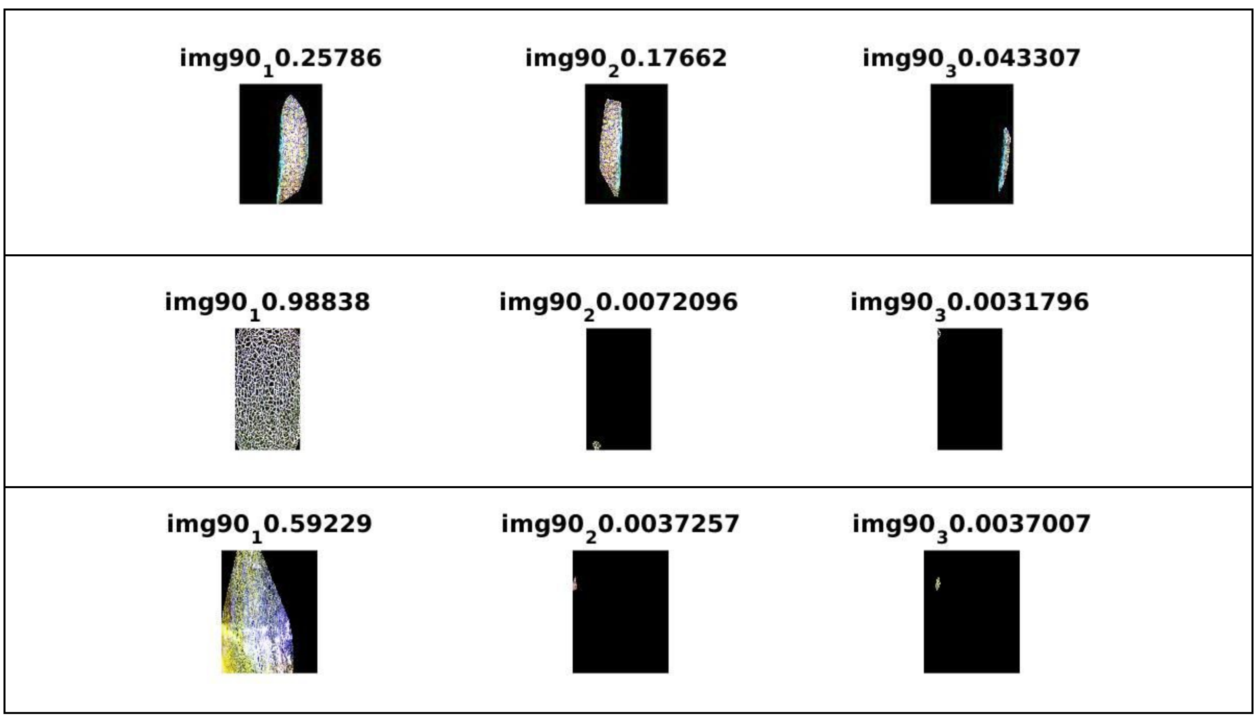

Fig.11 The regions in max rectangle and their area ratios.

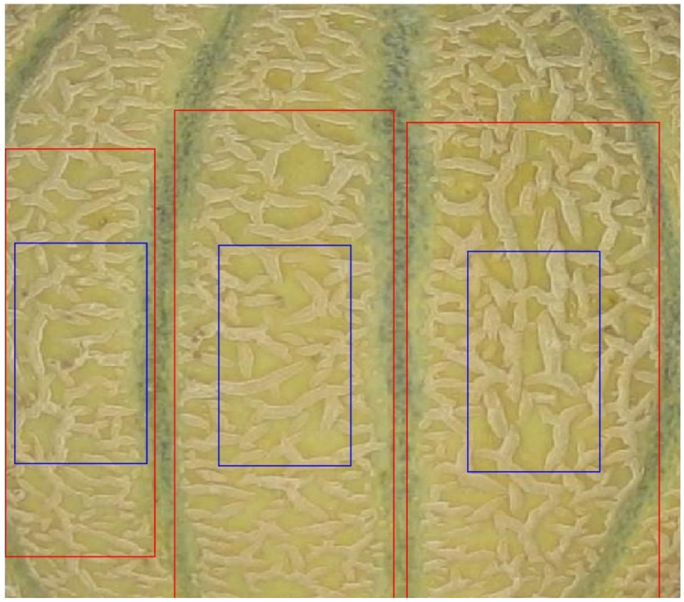

Fig.12 The region and cropped windows

Segmentation of netting from the melon background

The segmentation of the netting is one of the hardest tasks of this project. Many different approaches were tested, but no one performed well for all melons. One of the main difficulties is that the colour of the netting is not constant for all the melons - in fact the colour of netting changes a lot from melon to melon. This makes any approach of colour thresholding really complex. We developed some methods for thresholding based on HSV, setting different channels and different thresholds but we did not get good results for many melons. We also tried to do some pre-processing steps, such as histogram equalisation and sharpening of the edges, but our results did not improved.

Moreover, otsu thresholding was used in order to segment the method. It works well for many cases, but the problem is that we are not able to know what is the netting in the image. It turns out that, depending on the colour of the netting and the melon background, sometimes the threshold will represent the netting as black and sometimes as white.

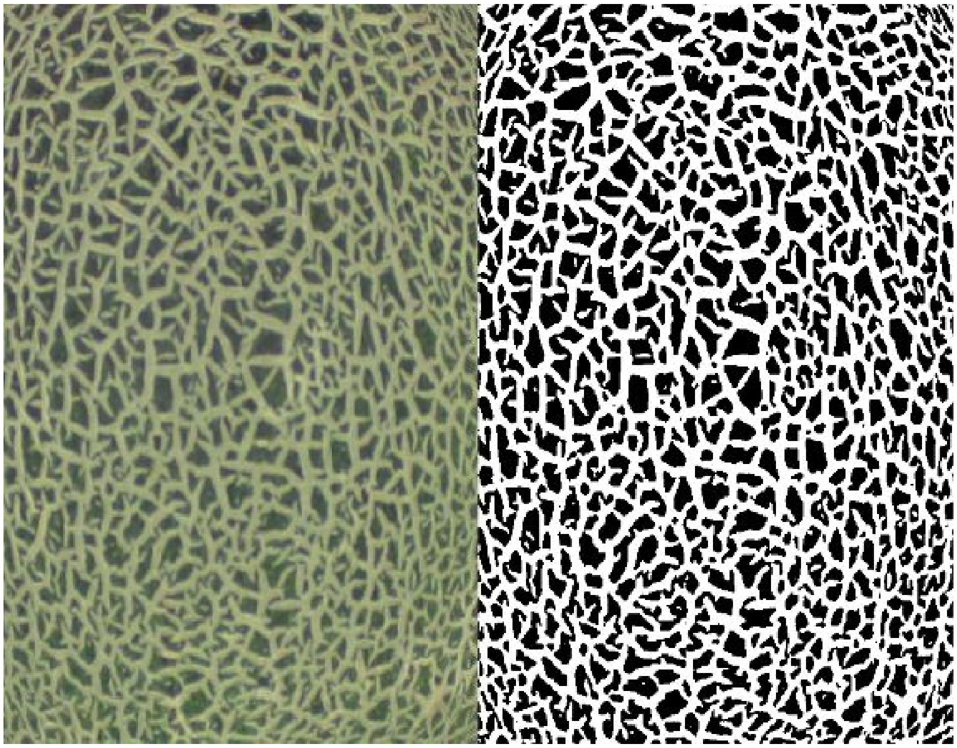

Figure 13: Melon segmented by Otsu Thresholding

Figure 13 and 14 are good examples of how the netting segmentation varies from case to case. In the figure 13 for example, the segmentation works really well, the netting is segmented by white colour. On the other hand, the figure 14 shows a bad segmentation of the netting, where the netting is represented as black.

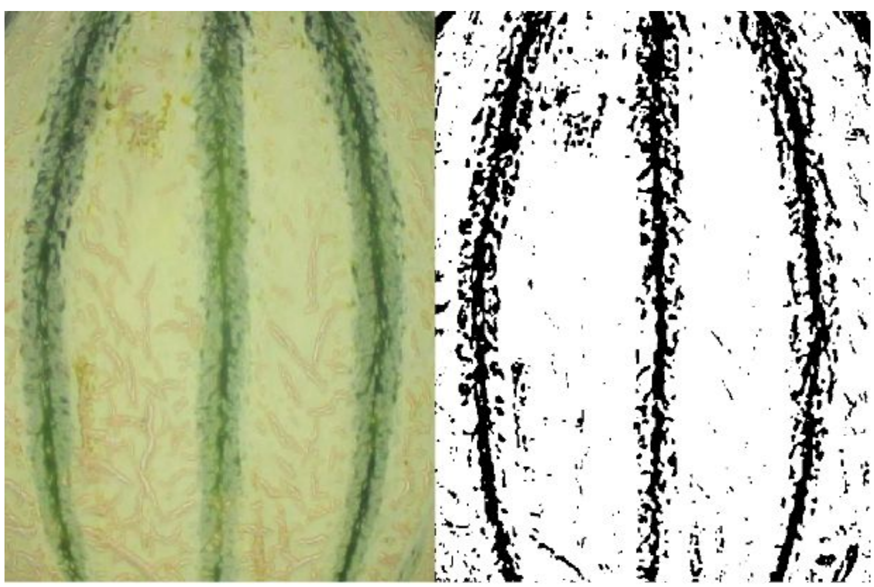

Figure 14: Melon segmented by Otsu Thresholding

In the previous part, we mention about the window approach to get improve the result of netting. Fig. 15 shows an example done with the window from fig. 14. As the dark bold green fringe is removed. The netting can be more visible. However, the scope of image is reduced.

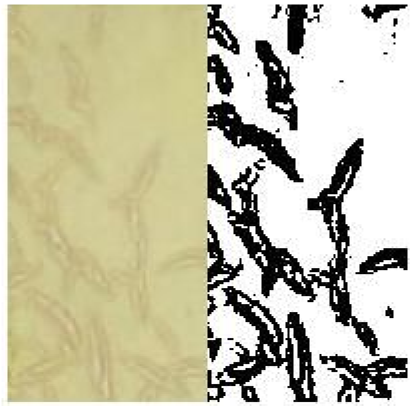

Fig. 15 Ostu Thresholding in center window. The window is from the fig.14.

We also tried k-means on the H and S channels of the HSV colour space. Using random seeds and two clusters for the k-means algorithm. Similarly to Otsu Thresholding, we get good results for some cases, and bad results for other cases. Furthermore, the problem of sometimes segmenting the netting as black and sometimes as white, which was described above the Otsu thresholding, can also be found for the k-means segmentation.

Another method which we tested for segmentation was edge filter, based on prewitt, sobel and canny. This method works well for uniform melons without stains and difference in colour in the background. However, when the melon has stains or difference in contrast in the background, the result of this method is not satisfactory.

Feature extraction and Machine learning

For the feature extraction we tried three methods: Gray-level co-occurrence matrix on all diagonals, up and down, left and right directions; harris corner detectors and Histogram of Gradients.

For the GLCM we use the contrast information in order to see if the contrast of the image is high. In theory, if the contrast is high, it means that the melon has dense netting; otherwise the netting is not so dense. However, this feature does not perform well because there are many melons with different contrasts and stains in the background. As a result, this method does not perform well for half of the cases.

By the harris corner detectors we saw that the number of corners is proportional to the number of netting when the melon is uniform. As the melons are not uniform for most of the cases, this feature also have problems.

In the histogram of gradients we were able to see that usually some classes of low dense nettings are directional, while high dense nettings are less directional. Similarly to the other methods described, it works well for some cases, but bad for other cases.

For the machine learning we used a simple k-nn classification algorithm, where we use 5 samples of each class for training, and the remaining for testing. The difficulty of using machine learning in this application is that there are not many samples for training. If we had a big diversity of samples, it would be possible to use neural network in order to do the classification. Another problem of using machine learning in this application is that we can not trust our ground truth, taking in consideration that different persons will grade the nettings of each melon in different ways. Moreover, in many cases, we can not tell the difference in netting between classes and they method for grading is not always coherent. All these mentioned problems make it really hard for using machine learning for classification.