Ferrero detector

Giudice Gianluca, Formenti Matteo

Macro step

- Box detection
- Vertices detection
- Box classification
- Perspective adjustment
- ROI Classification
- Error detection

BOX DETECTION

COLOR SPACE ANALYSIS

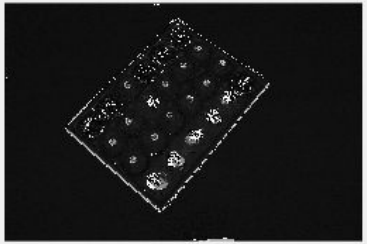
Color space analysis is critical in order to detect the most viable color space that maximizes the contrast between box and table. A qualitative analysis shows that Cb and Cr are the most appropriate channels.

RGB R

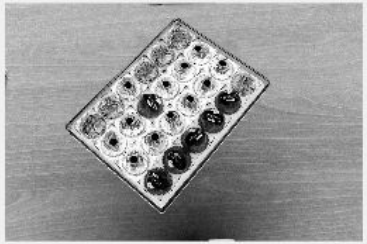
RGB G

RGB B

HSV H



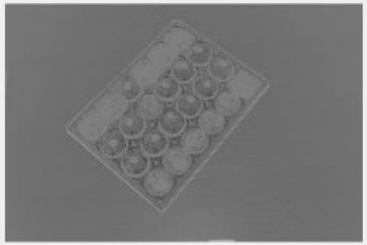
HSV S



HSV V

YCbCr Y

YCbCr Cb



YCbCr Cr

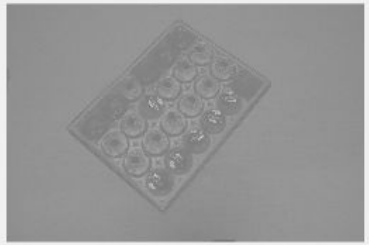
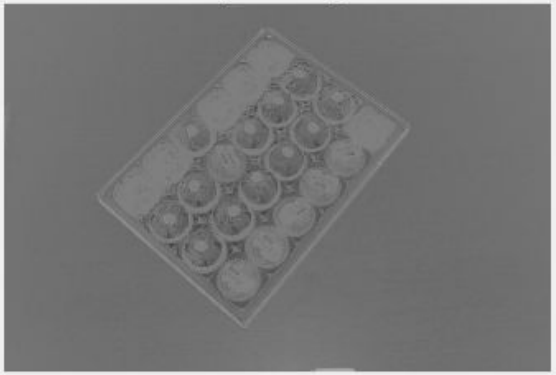


IMAGE ENHANCEMENT

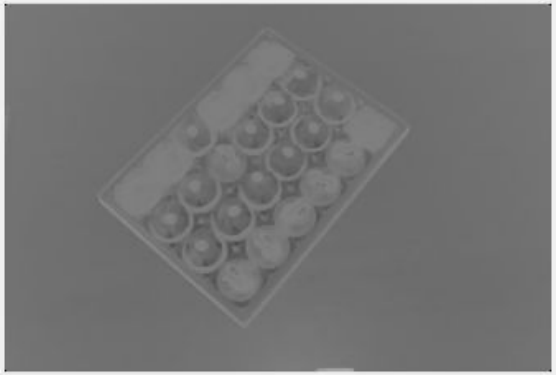
Filtering the image allows for higher box detection precision, the objective is to smooth as much as possible the background, so that edges are detected only inside the box. The following steps have been performed:

- Cb channel isolation
- Image resizing to a fixed dimension (allows for faster processing and static filter dimensions)
- Scaled image median and gaussian filtering

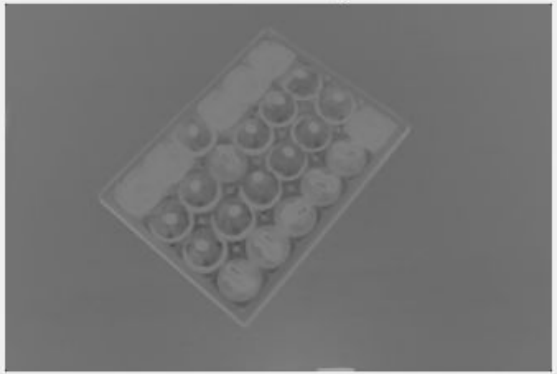
Input image



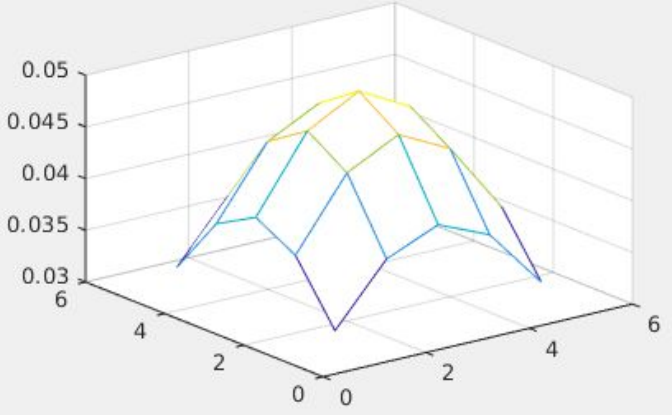
Median filter 15x15



Smoothing



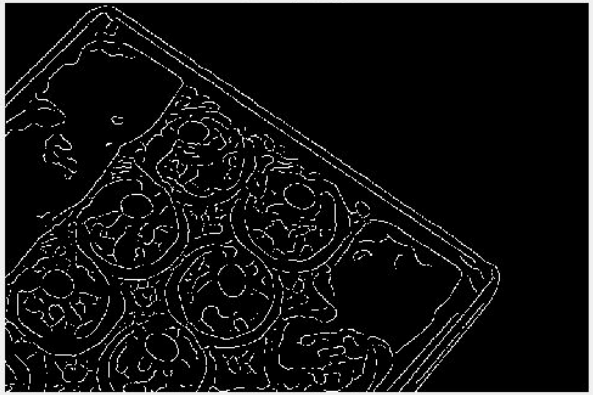
Gaussian filter (5x5, σ = 3)



EDGE DETECTION

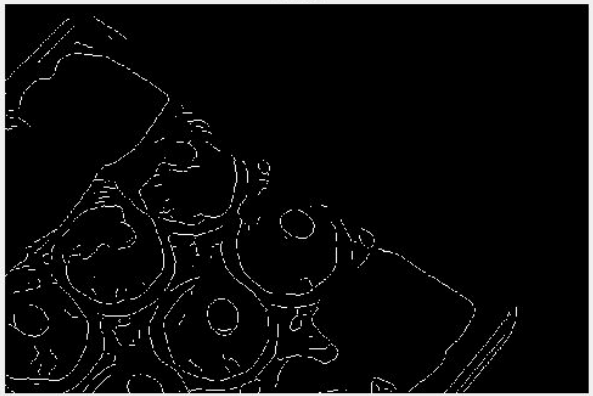
Canny edge detection returns a very “noisy” image in the box region (caused by the variance between all chocolates) and a somewhat clean image over the table. Using this image allows the algorithm to distinguish between box and table: high edge presence is classified as box or other objects, low edge presence is table.

Canny edge



Prewitt

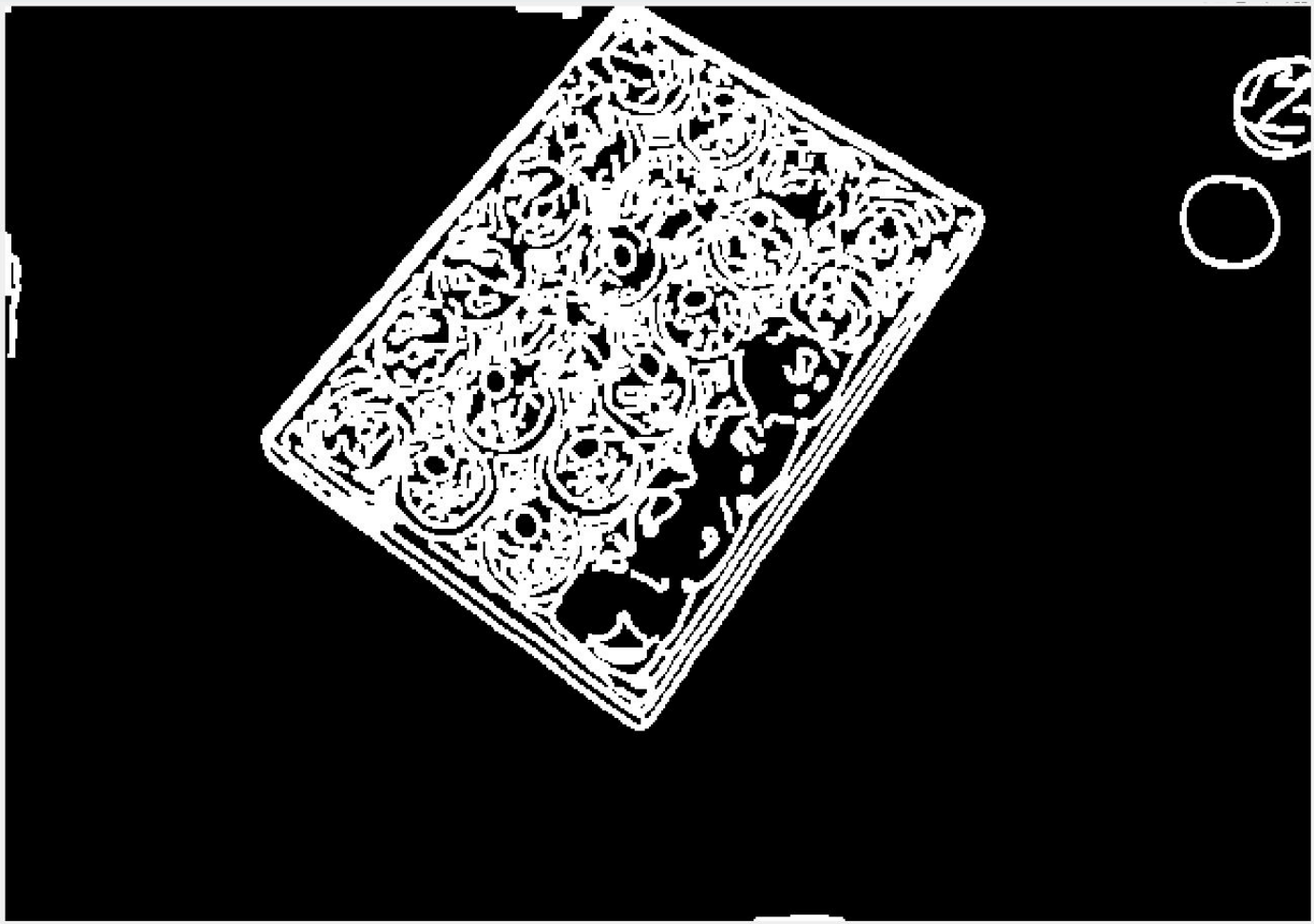
Sobel



Log

BOX DETECTION

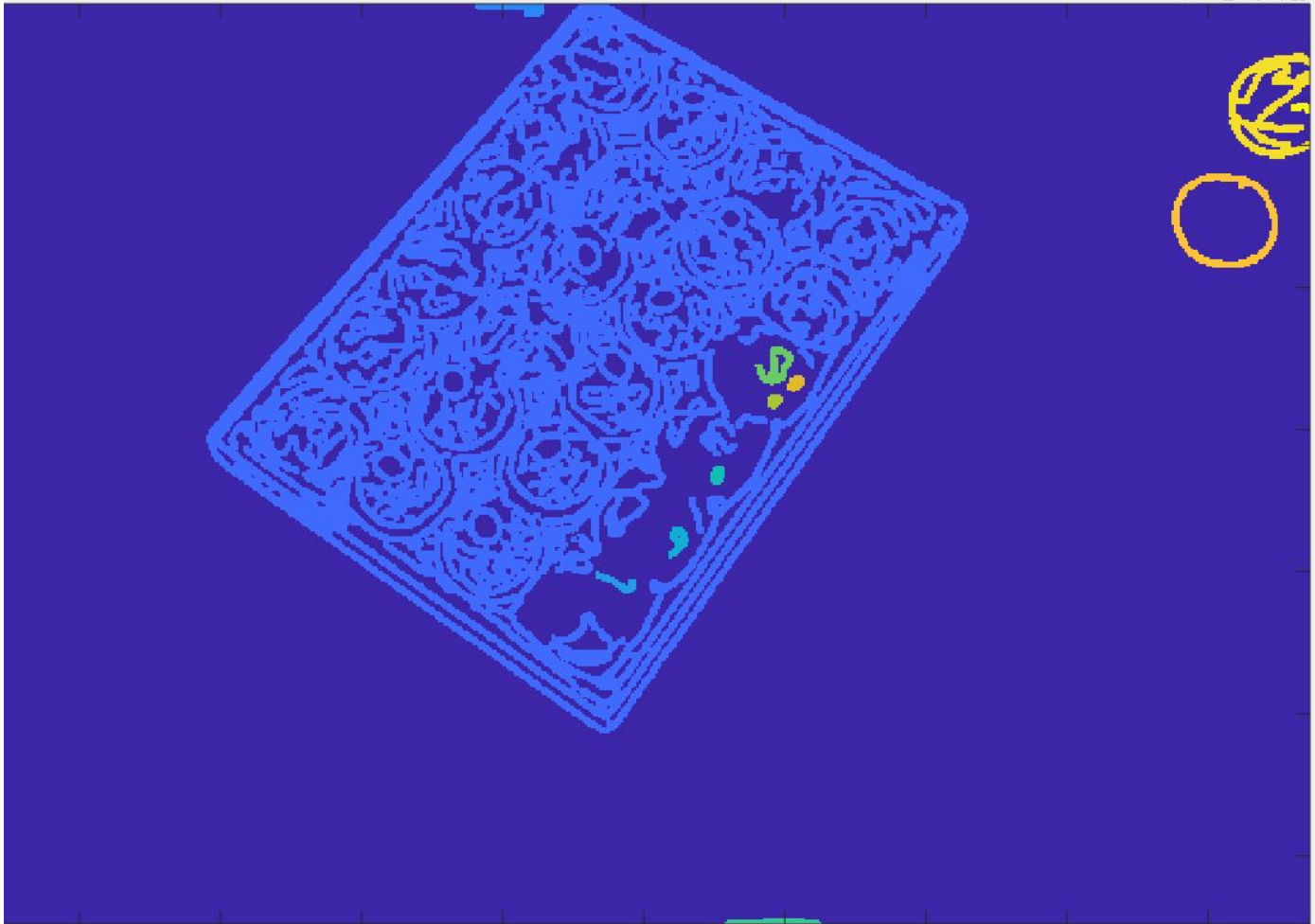
The edge image is binarized using a sliding mask of size ≈ ½ of the box edge, so that the transparent box is correctly classified. The image is considered box if the sum of all pixels beneath it is greater than 0.



OBJECT REMOVAL

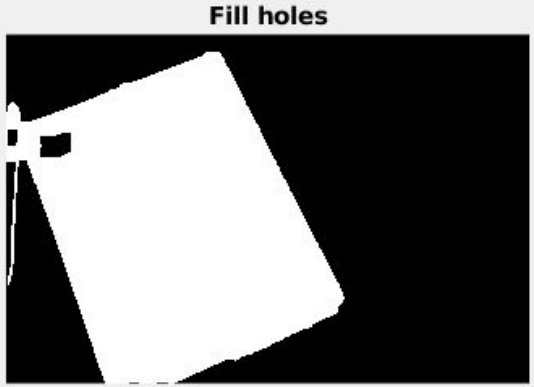
Connected regions labeling is performed on the binarized image, only the biggest component is maintained. Smaller regions are deleted (ie the two circles outside the box)

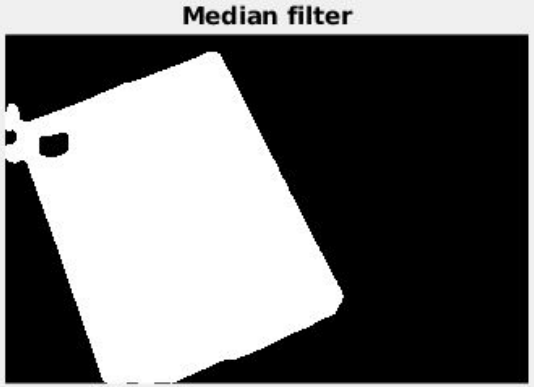
ASSUMPTION: the box is the largest component of the image

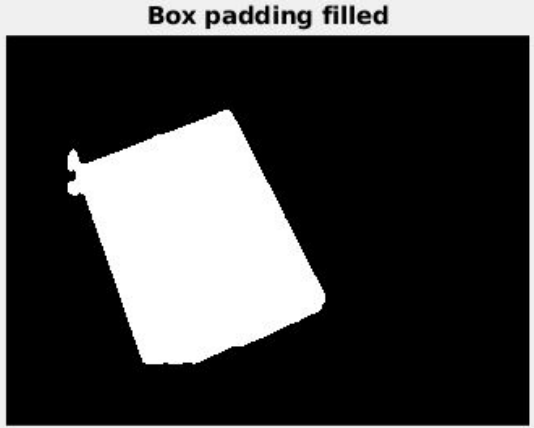


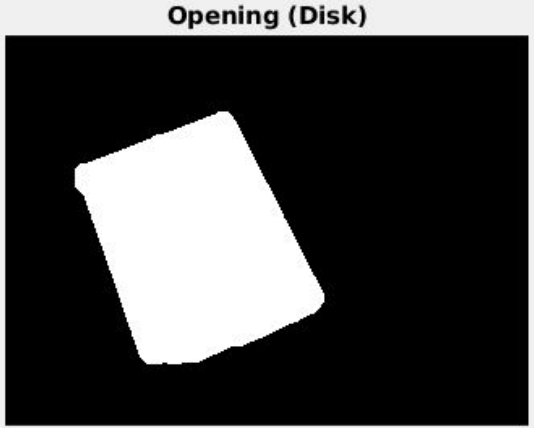
SHAPE CLEANUP

Holes, bulges and recesses are removed using mathematical morphology. This step is critical for the next one.







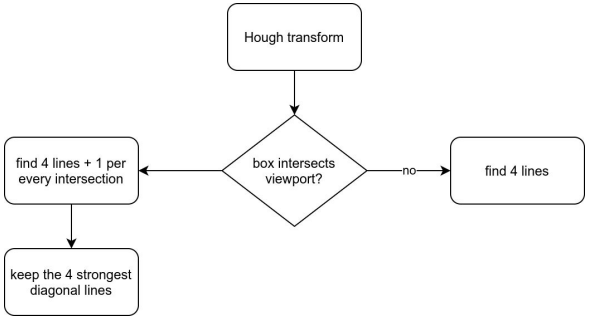


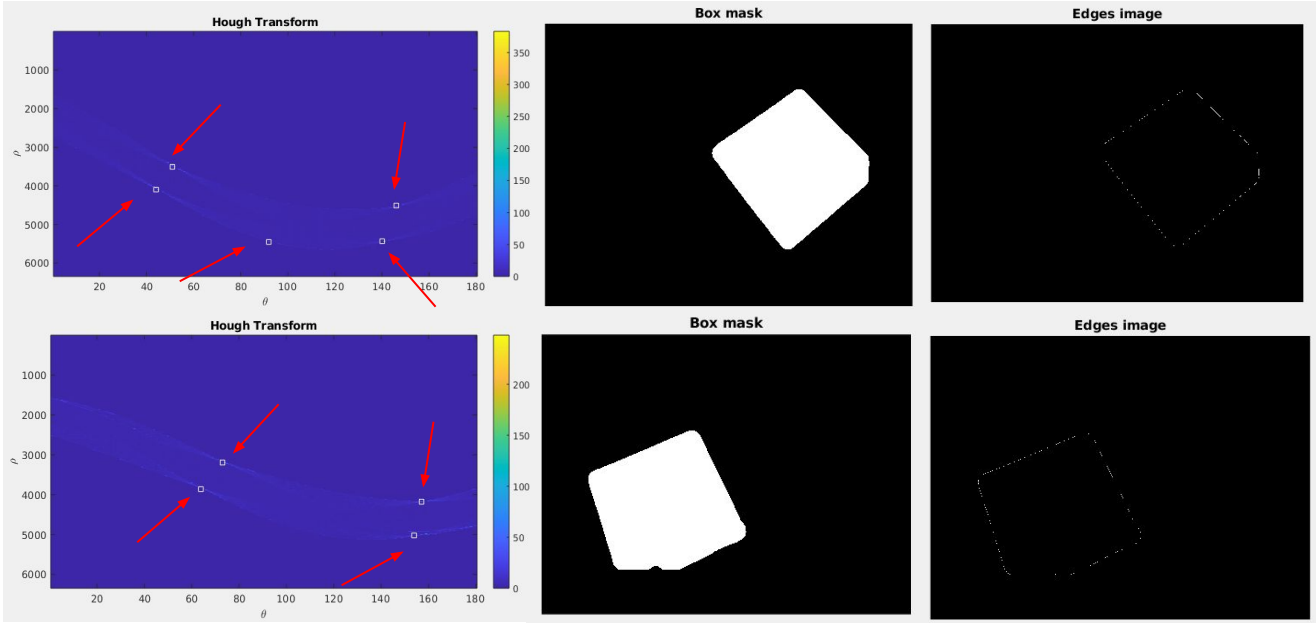
VERTICES DETECTION

HOUGH TRANSFORM

Hough transform is performed on the image to detect the lines that compose the box. By finding the correct number of lines it's possible to detect all edges, even the ones outside the image.

If the box intersects with the image border, it should not be parallel with any side of the image



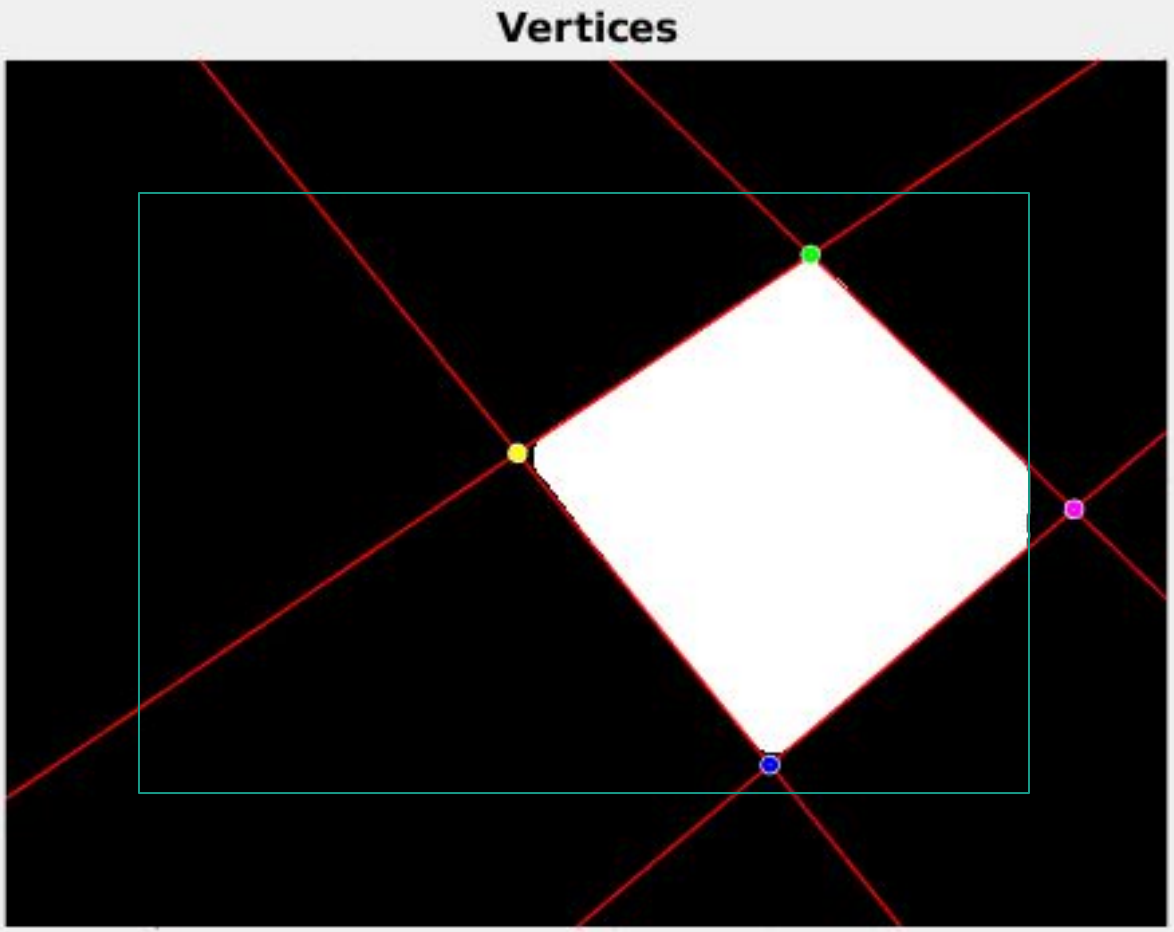


VERTICES DETECTION

The detected 4 lines are then used to calculate 4 vertices inside of the padded viewport. The green region shown is the image, some adding is added to allow for vertices outside the original image.

All 4 vertices can be outside of the image and this method still can find them. Vertices outside the padded image aren't calculated since those are the focal points of the perspective.

Convex hull is then calculated to sort the vertices, then the list is sorted so that the first two points make up the longest segment



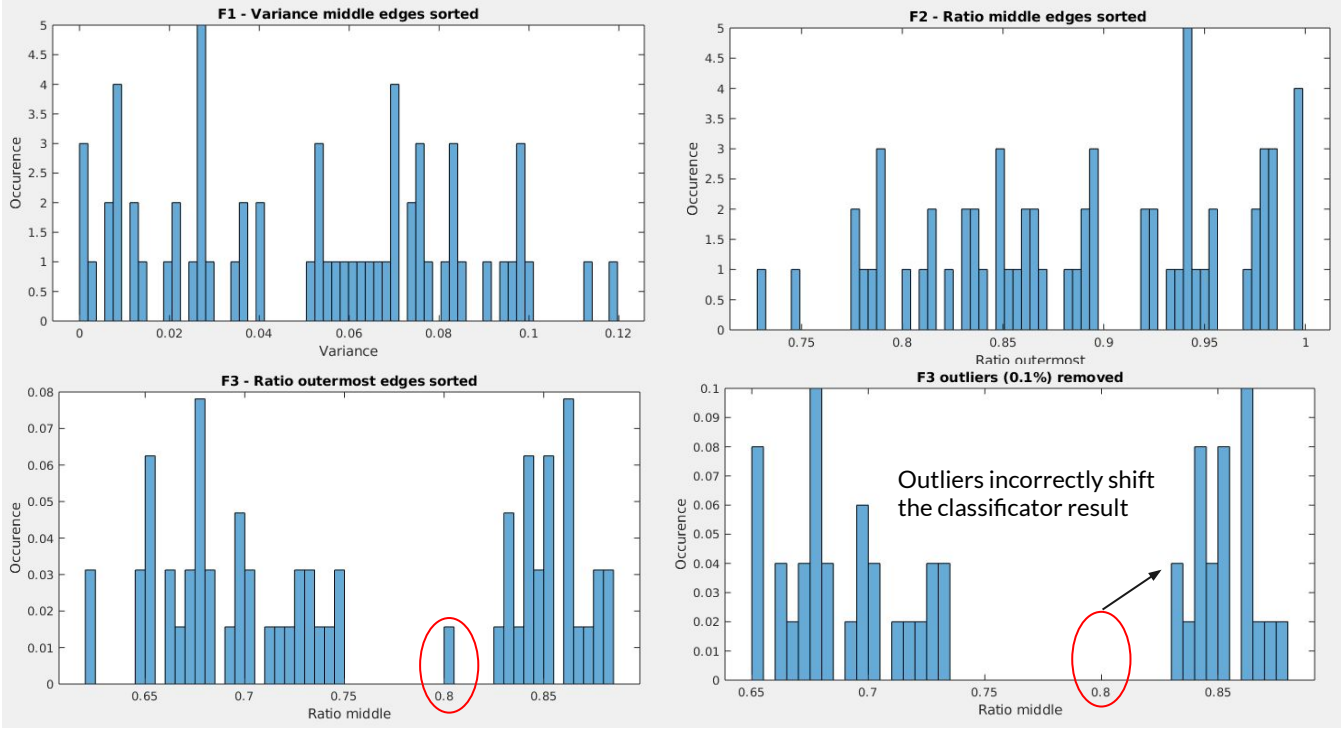
BOX TYPE CLASSIFICATION

BOX TYPE CLASSIFICATION

The length of each side is computed, outliers are removed and using the ratio of the longest and the shortest side, the box is classified as square or rectangle.

Three features have been considered for the classification, F4 is the most accurate one

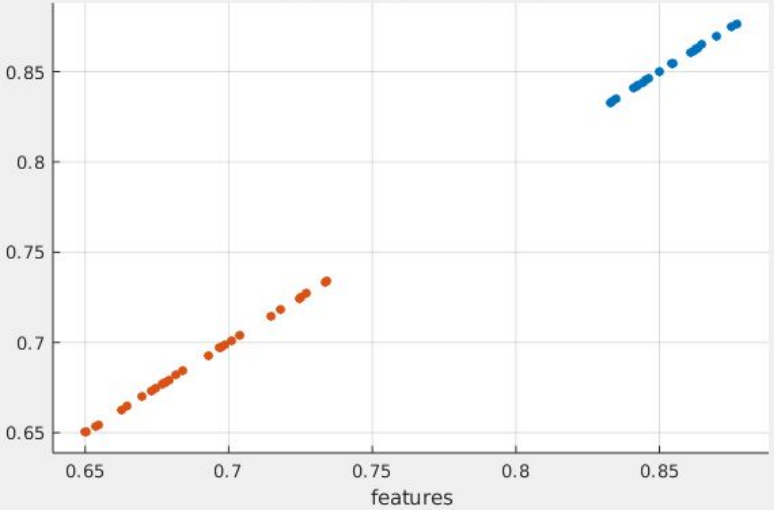
1. **F1**: Variance between medians
2. **F2**: Length ratio of medians
3. **F3**: Length ratio between longest and shortest
4. **F4:** F3 without outliers

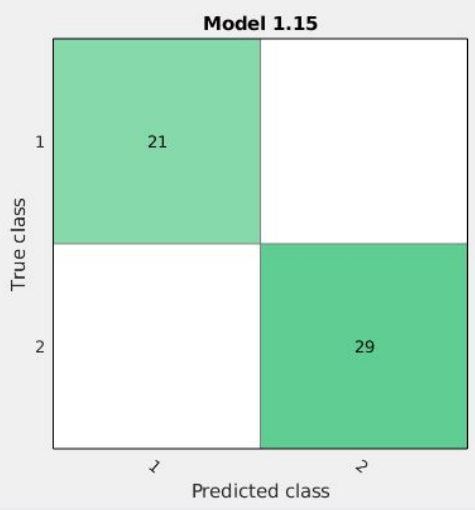


CLASSIFICATOR PERFORMANCES

Class 1 = square box
Class 2 = rectangle box
Classifier: **1-NN**
6-fold cross-validation

ACCURACY = 100%
ACCURACY(with outlier) = 97%





6-fold cross-validation

PERSPECTIVE ADJUSTMENT

PERSPECTIVE ADJUSTMENT

After sorting all 4 vertices in order to have the longest edge first in the list, there vertices are then mapped into 4 fixed points (whose position is based on the box type).

The resulting image can then be cropped. Removing some padding from all sides simplifies the next steps since the calculated vertices are always outside the real box.

Original Image

Perspective adjust

Box cropped

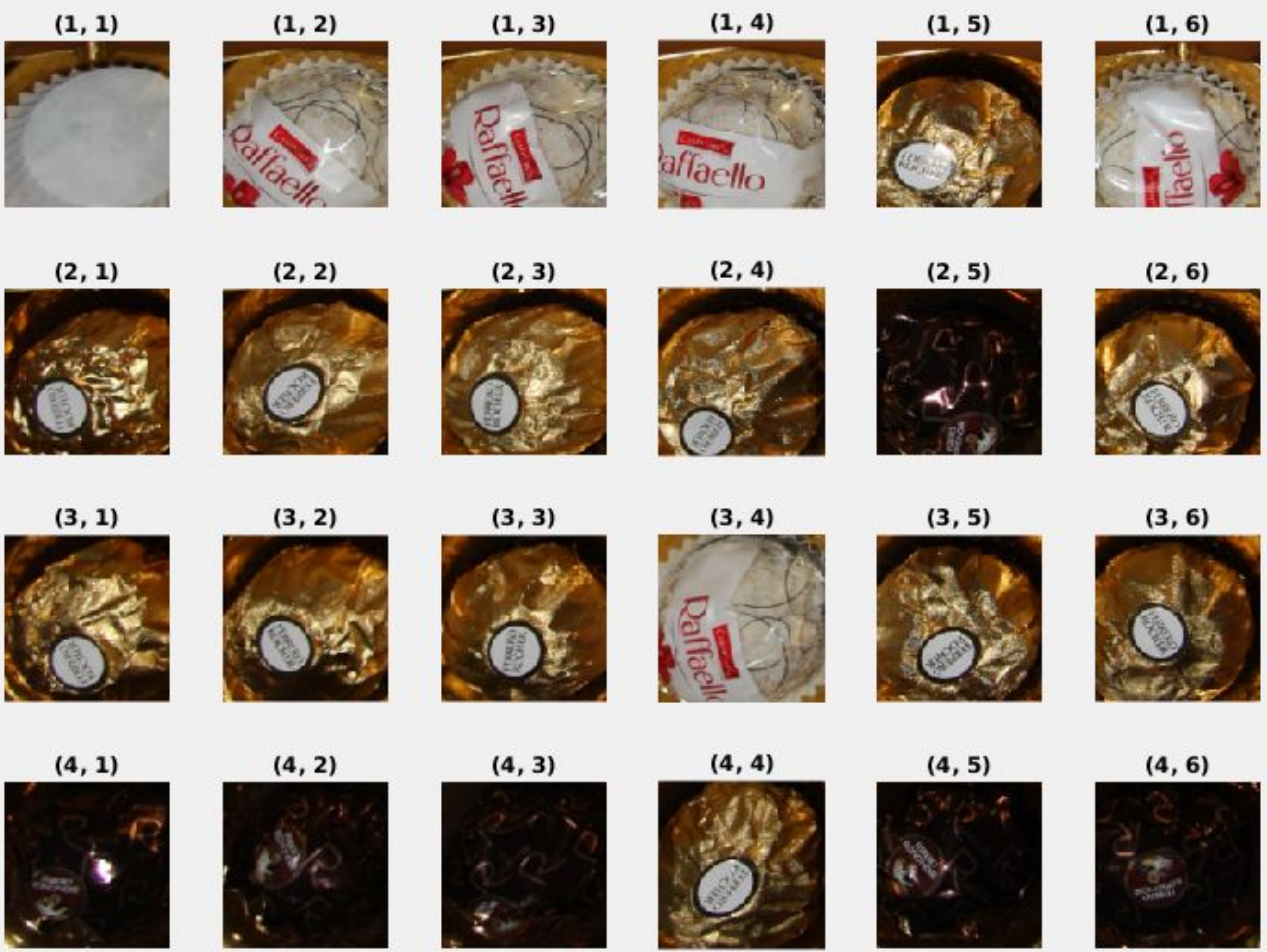
Crop enhanced

ROI CROPPING

ROI CROPPING

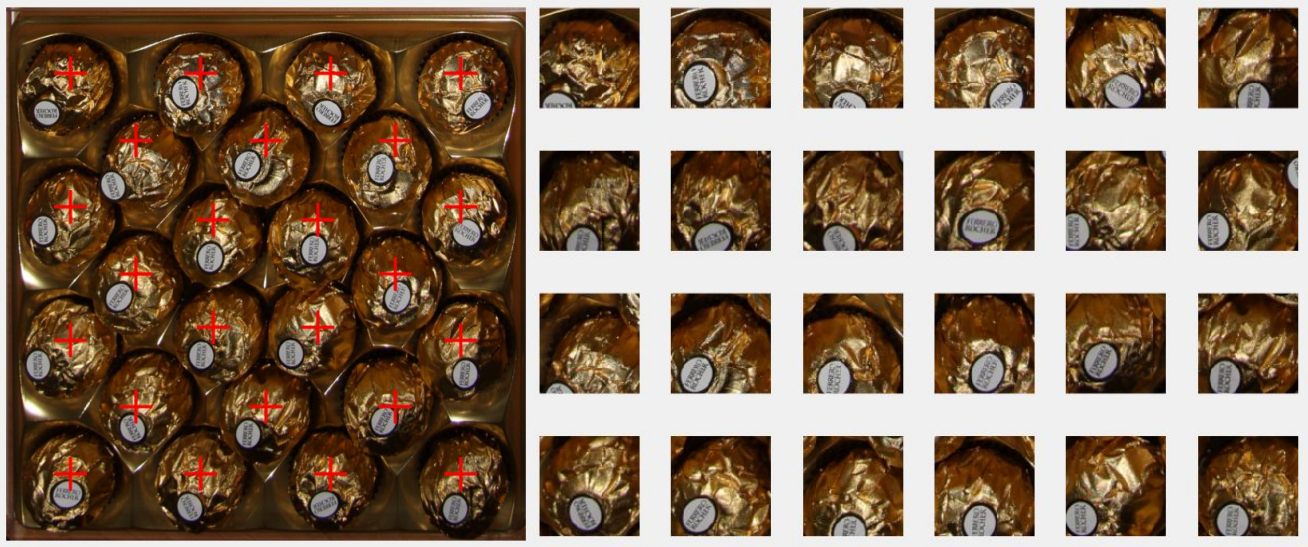
Having the perspective correctly adjusted allow the algorithm to cut all chocolates precisely without additional processing.

A sample result is shown



ROI CROPPING

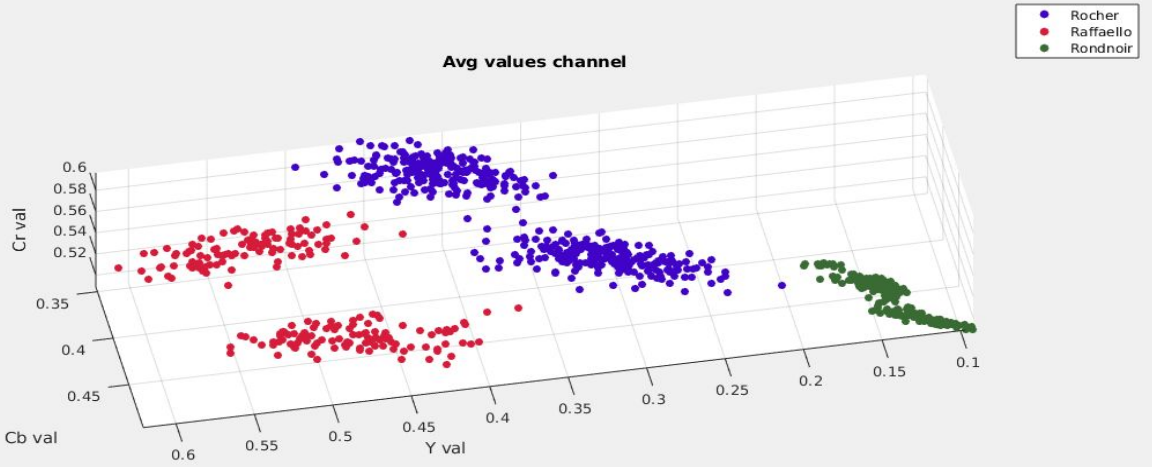
The developed function allow for simple edit of cut parameters like radius and shape.

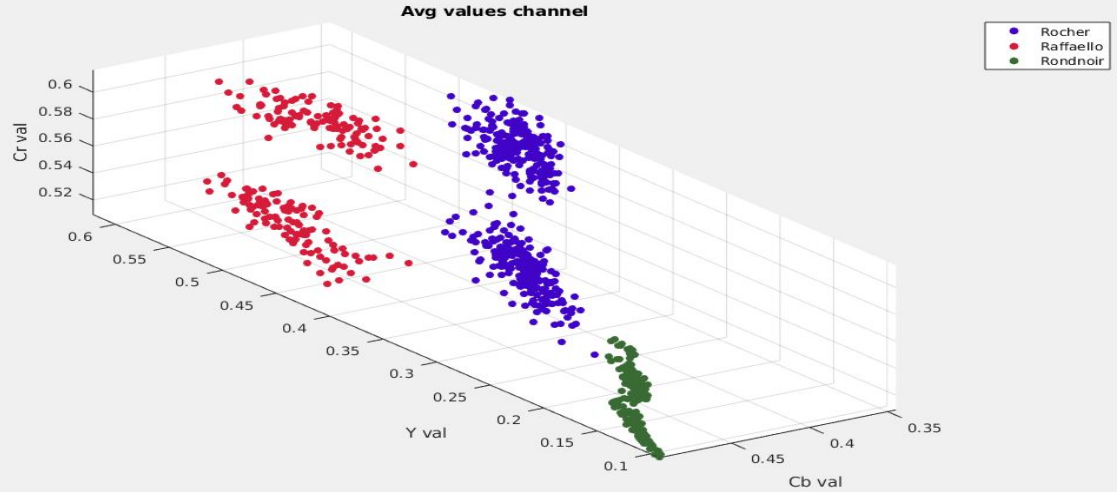


CLASSIFICATION FEATURES

AVG YCbCr

YCbCr, average among all channels





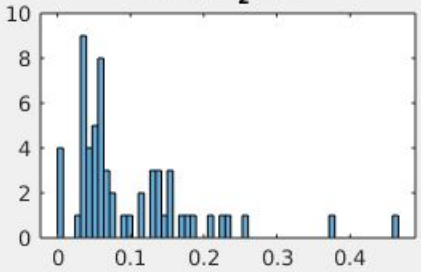
LBP

Rocher$_1$ RGB

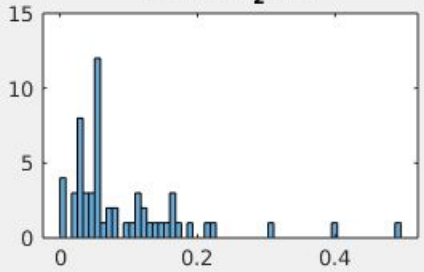
Raffaello$_1$ RGB

Rondnoir$_1$ RGB

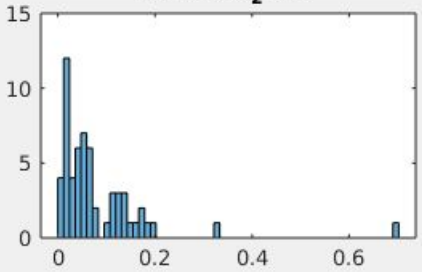
Rocher$_2$ LBP



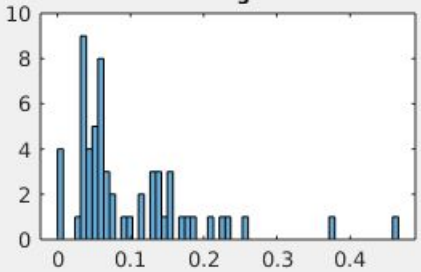
Raffaello$_2$ LBP



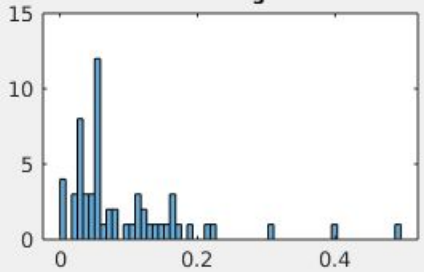
Rondnoir$_2$ LBP



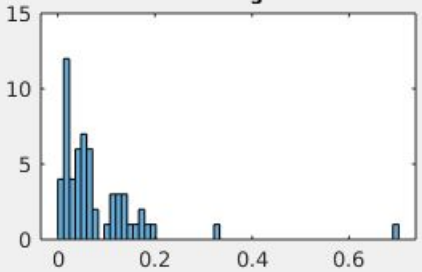
Rocher$_3$ LBP



Raffaello$_3$ LBP

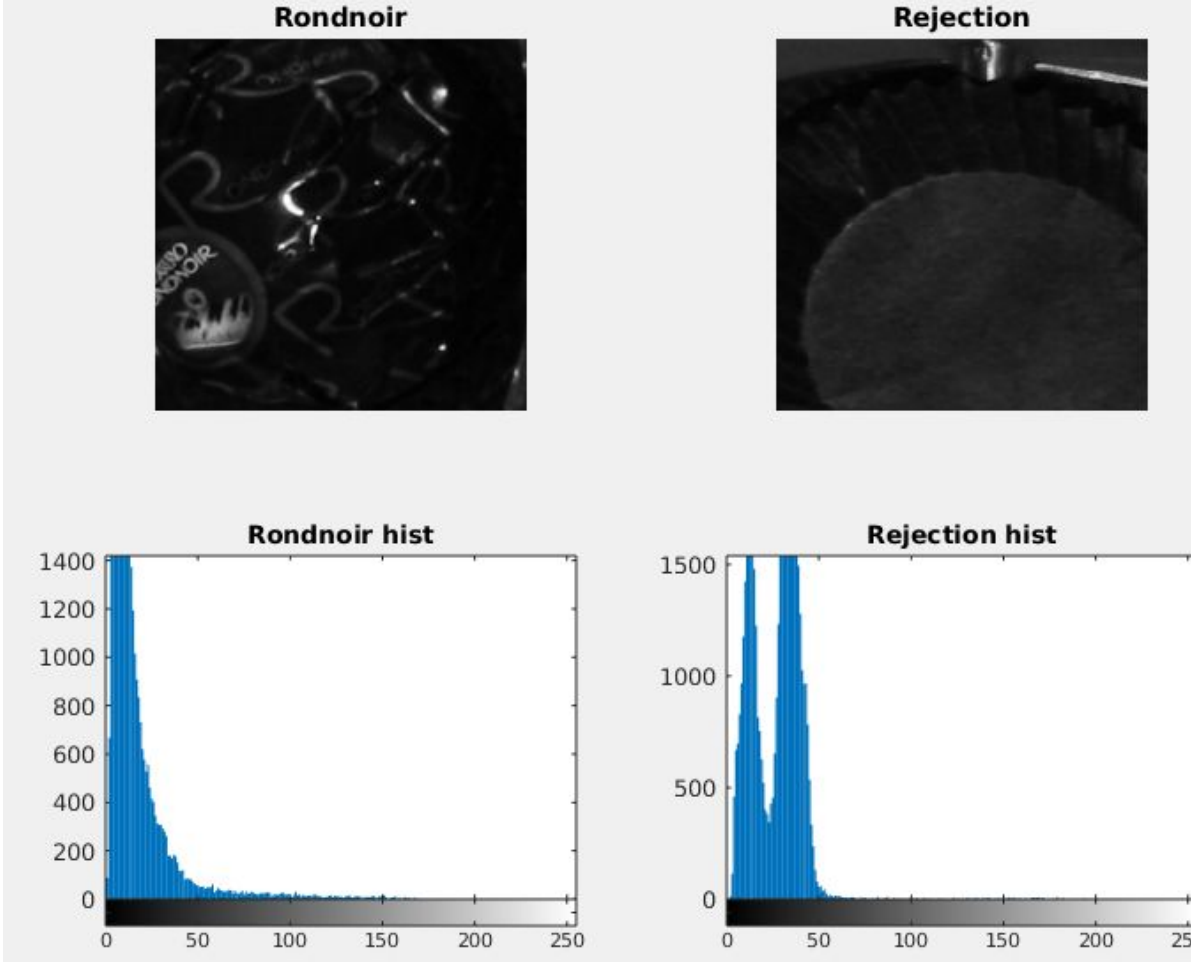


Rondnoir$_3$ LBP



STD

Standard deviation of cuts converted in grayscale image



CLASSIFIER PERFORMANCE

DATASET PREPARATION

A CSV file is manually populated with 888 ROI name and class, classes are defined below

1. Rocher
2. Raffaello
3. Rondnoir
4. Reject

ROIs have been automatically generated using previous steps, so the classification is computed on actual images.

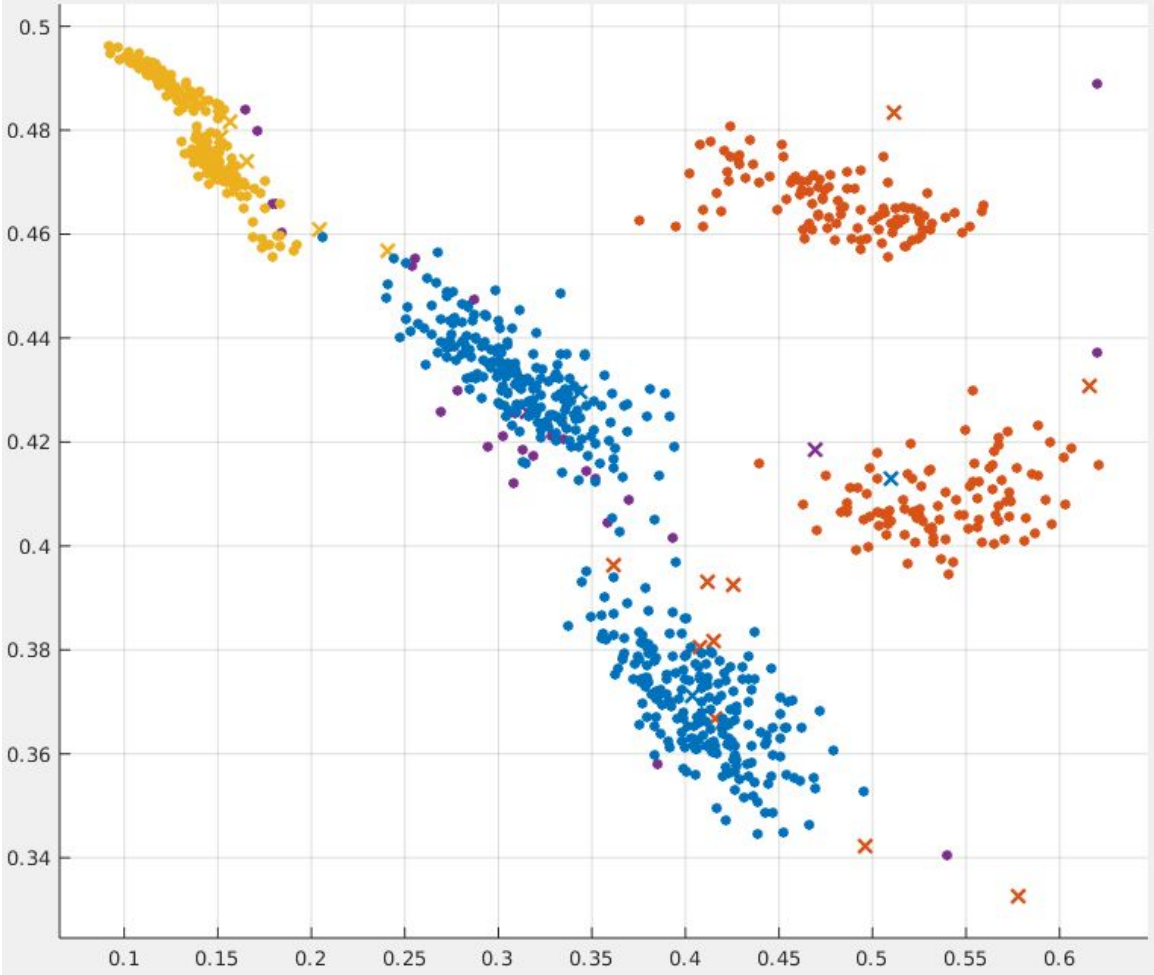
```
Cut,Label
IMG_10-10.png,1
IMG_10-11.png,1
IMG_10-12.png,1
IMG_10-13.png,1
IMG_10-14.png,1
IMG_10-15.png,1
IMG_10-16.png,1
IMG_10-17.png,1
IMG_10-18.png,1
IMG_10-7.png,1
IMG_10-8.png,1
IMG_10-9.png,1
IMG_11-10.png,1
IMG_11-11.png,1
IMG_11-12.png,1
IMG_11-13.png,1
IMG_11-14.png,1
IMG_11-15.png,1
IMG_11-16.png,1
IMG_11-17.png,1
IMG_11-18.png,1
IMG_11-7.png,1
IMG_11-8.png,1
IMG_11-9.png,1
IMG_12-10.png,1
IMG_12-11.png,1
IMG_12-12.png,1
IMG_12-13.png,1
IMG_12-14.png,1
IMG_12-15.png,1
IMG_12-16.png,1
IMG_12-17.png,1
IMG_12-18.png,1
IMG_12-7.png,1
IMG_12-8.png,1
IMG_12-9.png,1
IMG_13-10.png,1
IMG_13-11.png,1
IMG_13-12.png,1
IMG_13-13.png,1
IMG_13-14.png,1
IMG_13-15.png,1
IMG_13-16.png,1
IMG_13-17.png,1
IMG_13-18.png,1
IMG_13-7.png,1
```

IMG_10-10.png, class: Rocher

FEATURES ANALYSIS

- Classifier type: **1-NN**
- Features: **AVG+LBP**
- 6-folds validation

ACCURACY = 97.6%

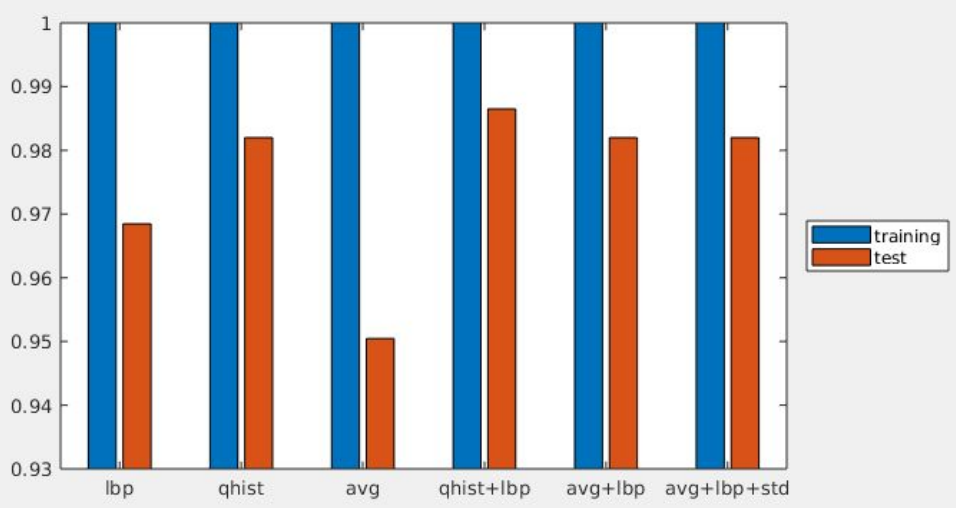


TEST MULTIPLE FEATURES COMBINED

1-NN Classifier

6-fold cross-validation

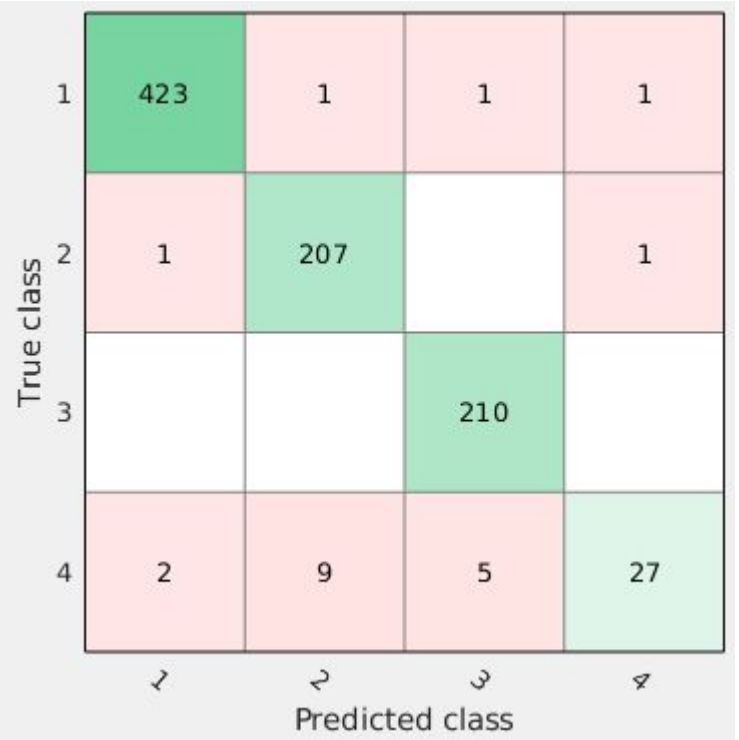
Accuracy



CONFUSION MATRIX

Most of misclassifications are about rejection class, which are classified as valid chocolates

Since only valid rocher have tags on it, we introduce a new feature in order to achieve a better accuracy.



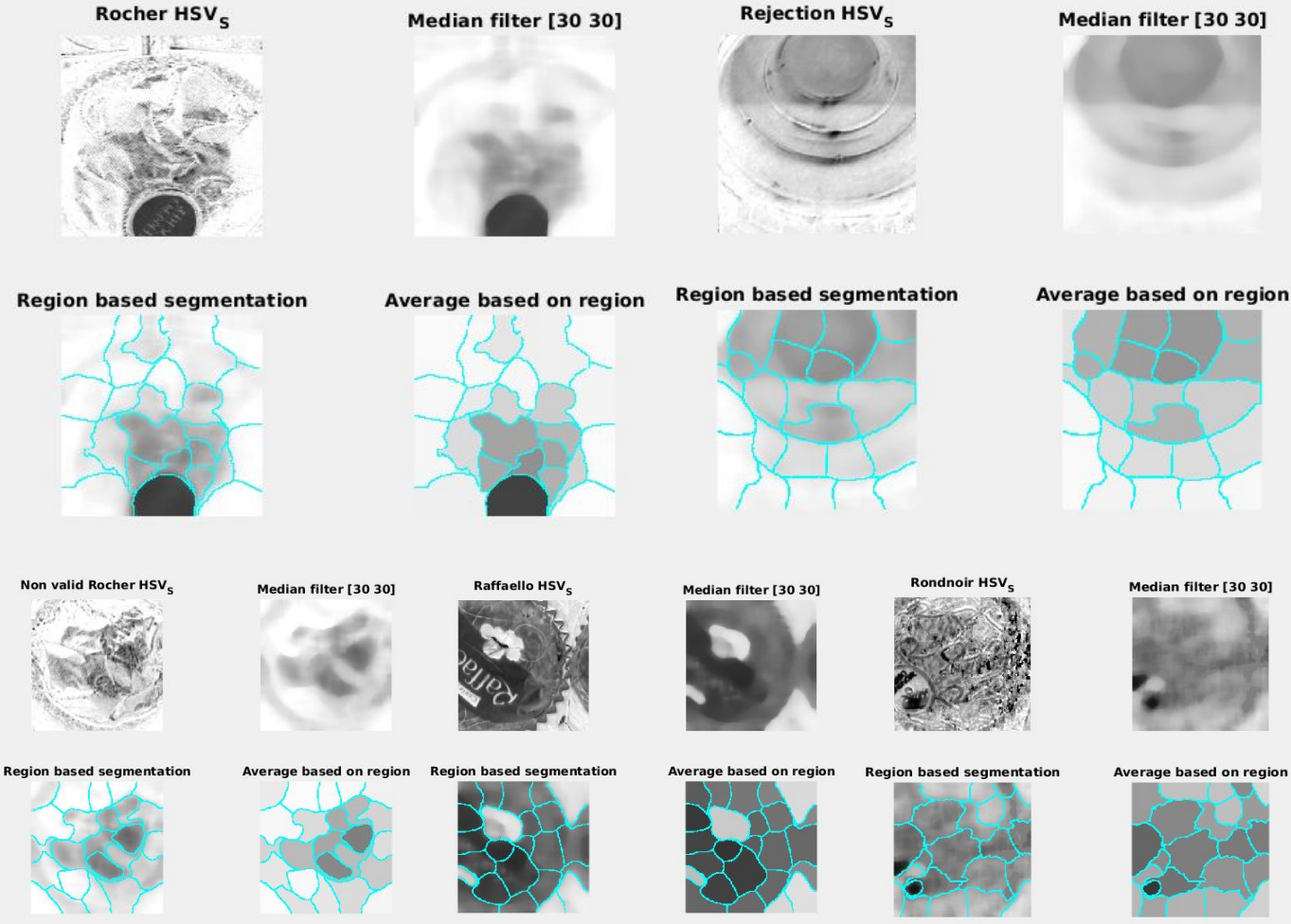
NEW FEATURE

The feature consists in a region based segmentation. Regions are recursively splitted and then merged based on near regions with low variance. Eventually the average value of each region is computed and the lowest value is returned.

This new feature is useful for detect valid rocher vs rocher without tag. Also the classification between rocher and rejection class is easier.

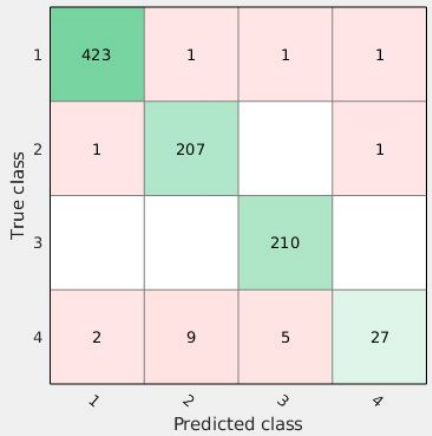
Even if rocher, raffaello and rondnoir are not distinguishable using this feature, the avg feature is very different

Therefore there should be an overall improvement in classification

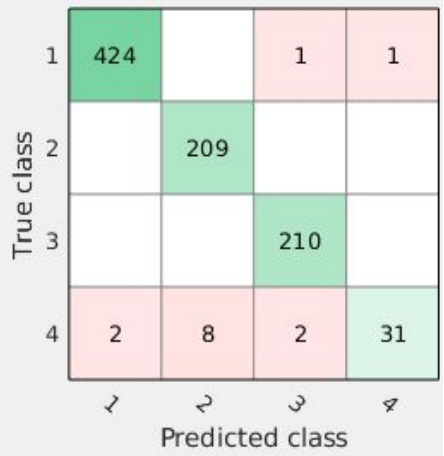


PERFORMANCE IMPROVEMENT

- Classifier type: **1-NN**
- Features: **AVG+LBP**
- 6-folds validation



New feature



ACCURACY = 97.6%

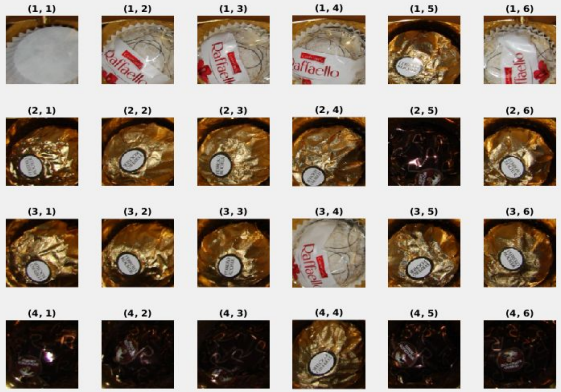
Improvement

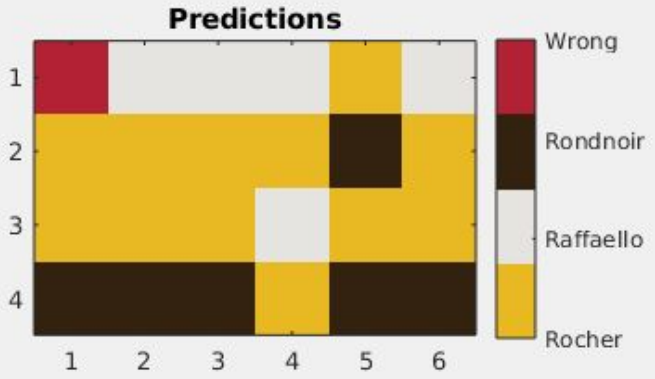
ACCURACY = 98.4%

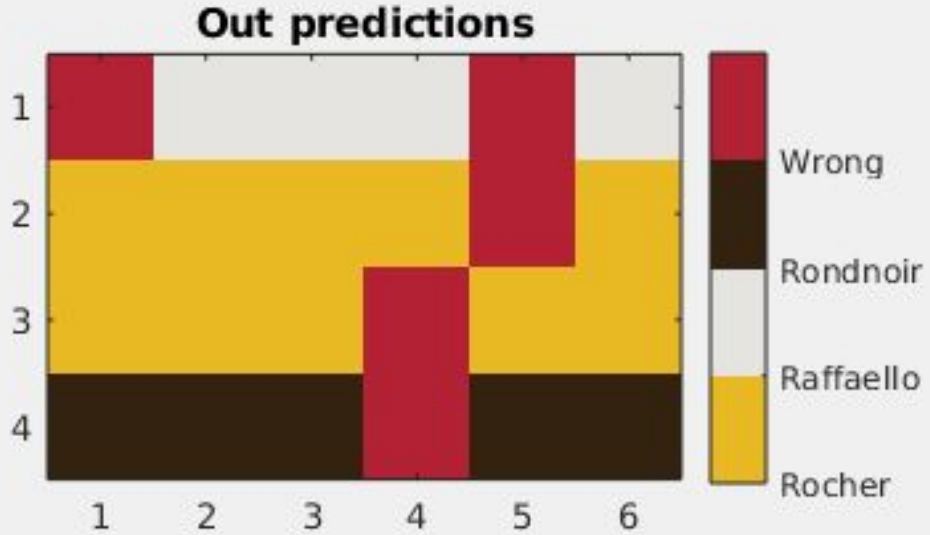
CUTS CLASSIFICATION

ERROR DETECTION

An image mask is compared with static validity masks (that are different for each box type), no difference means the input image correctly matches a reference matrix, the box is correct.







RESULTS

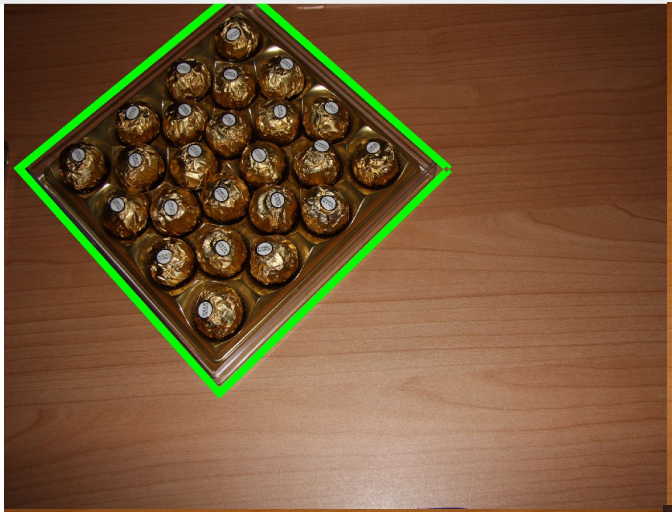
RESULTS

After classification, all invalid results are transformed back to the original image and displayed in the output as blue circles.

Total boxes: 64
Correct boxes: 38
Incorrect boxes: 26

Correctly defined as compliant	30/38
Incorrectly defined as non-compliant	8/38
Correctly defined as non-compliant	24/26
Incorrectly defined as compliant	2/26

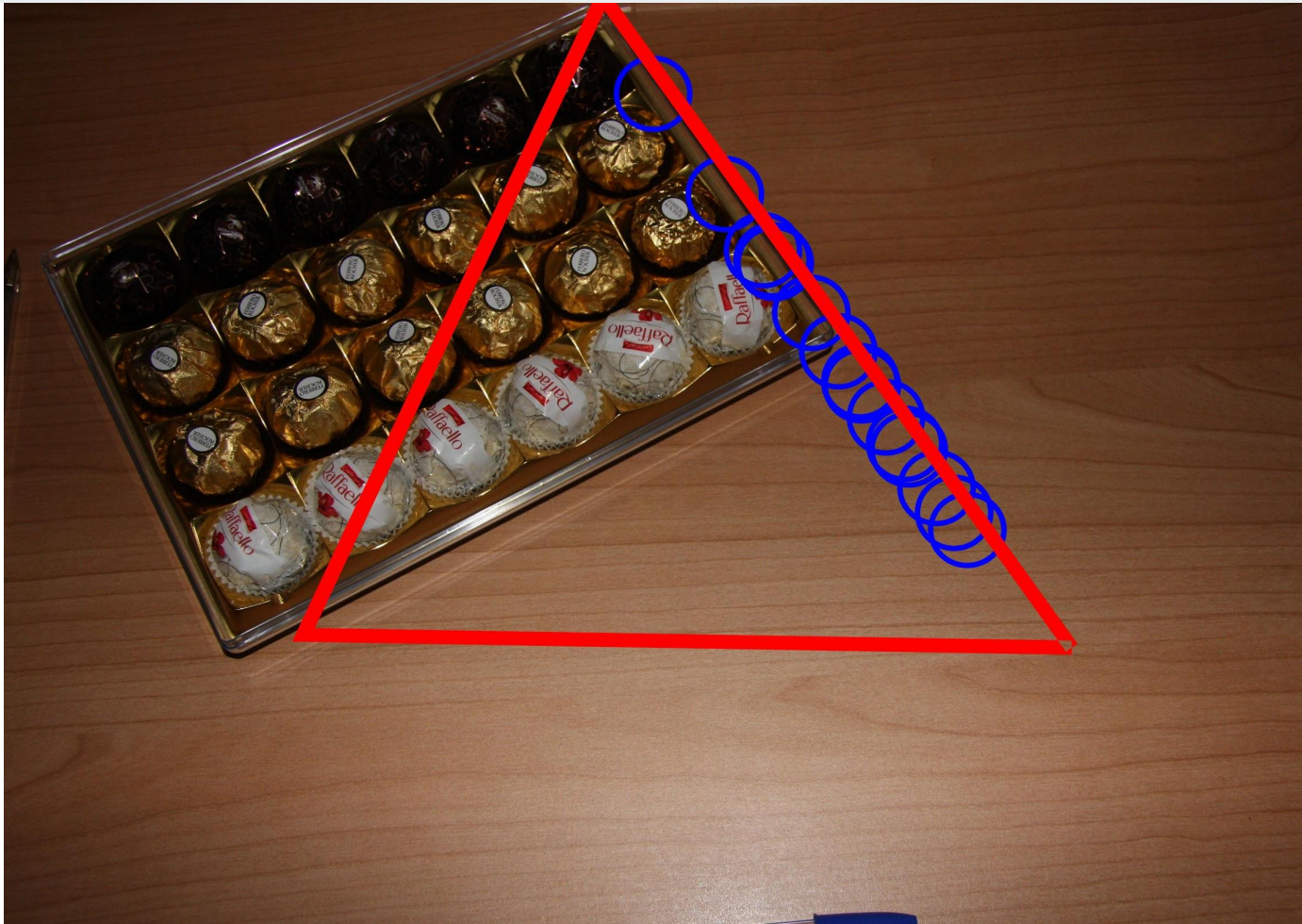
Overall accuracy: 84.37 %



COMMENTS

Some errors are caused by bugs in the box detection algorithm, when the contrast of the wood venis is too high, edges unuseful are detected and the result hough transform doesn't match the real box. This results in a false positive rejection class classification.

The key for improving this kind of errors is to strengthen the box detection algorithm, either by using more complex segmentation algorithm (combining various edge detection methods, pixel based classification for table/non-table, general table/non-table classification) or by improving the existing solution with more advanced filtering, morphology correction or by combining different color spaces/pipelines



COMMENTS

Dynamic ROI cropping is probably the second most impactful upgrade that could be developed in order to increase the overall accuracy. Even though the perspective is corrected in a previous step, the precision is still somewhat imprecise, and the percentage based cutting that is currently implemented could lead to some miscalculation.

Many tests have been performed to improve the crop quality, the most successful (but still not valid) have been:

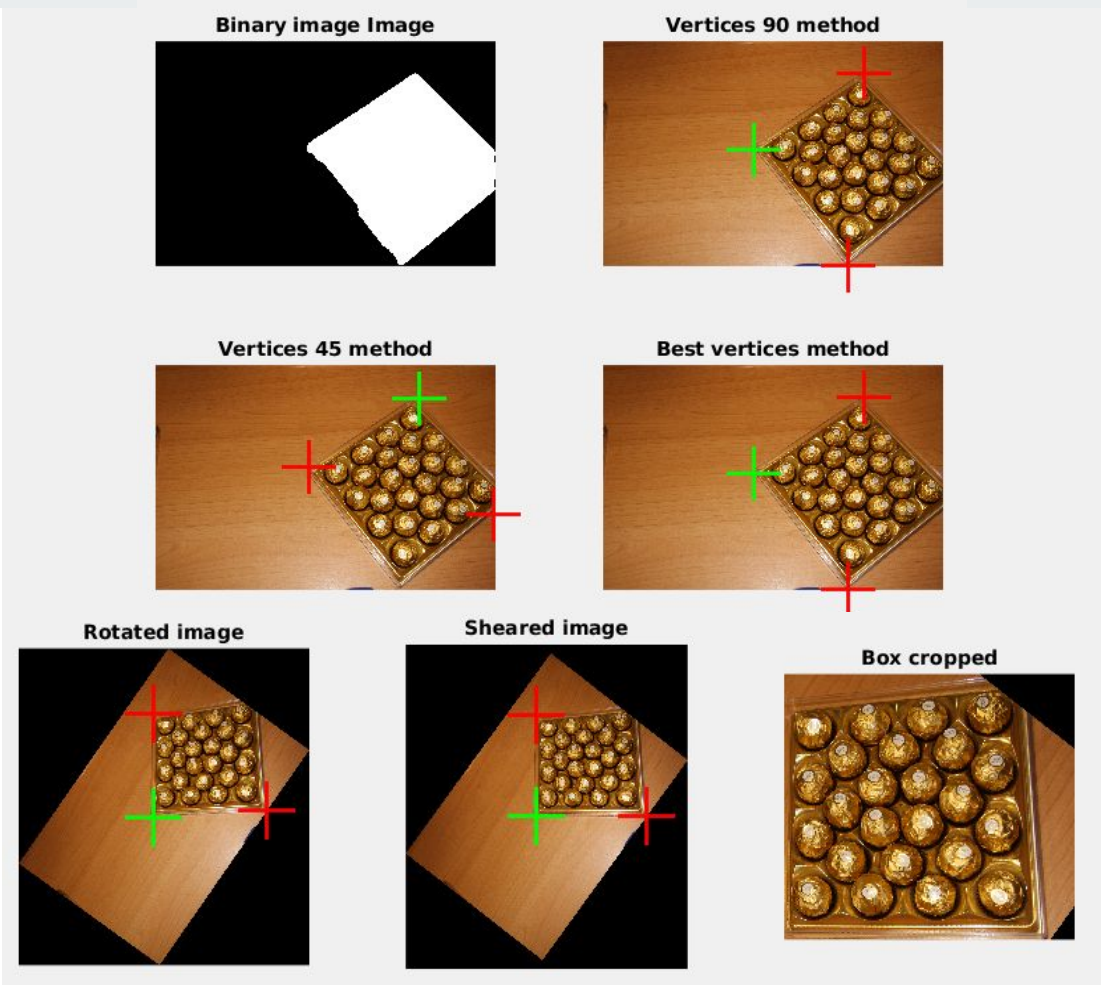
1. Using superpixel oversegmentation to more accurately cut ROIs. Fails because the color between rocher and the box is too similar, the number of superpixel cannot be calculated.
2. Edge detection and hough analysis to detect line patterns, valid idea but infeasible, edge detection is not accurate enough
3. Pattern matching, computationally infeasible, too many versions of each chocolate have to be computed

FAILED TESTS

VERTICES DETECTION

Detecting vertices is done by sliding a virtual line diagonally and vertically, then the method that gives the most accurate result is selected automatically. Only 3 vertices can be detected when the box is outside the viewport.

Using the detected vertices is possible to rotate, shear and crop the box using linear transformations.

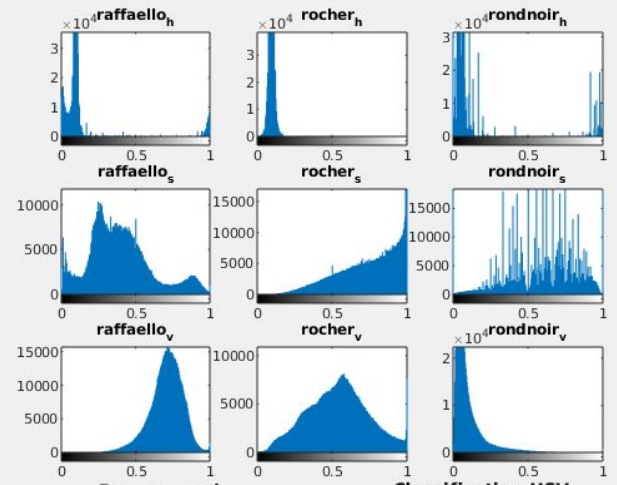


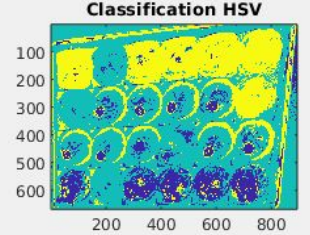
PIXEL BASED CLASSIFICATION

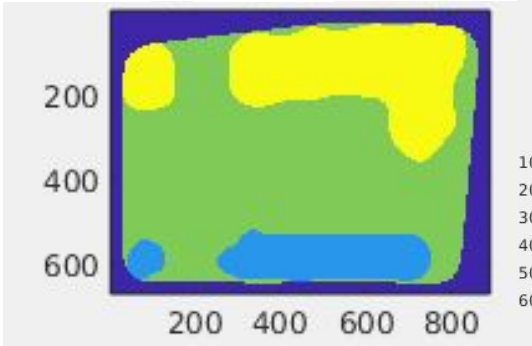
As for the first step, qualitative color space analysis is performed on a sample of various classes. HSV channels shows a clear distinction between the three classes. Mathematical morphology is then performed on the classificator result to separate the various ROIs.

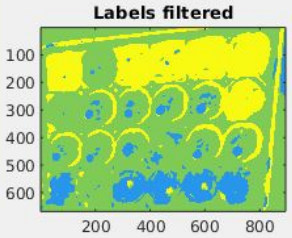
Mathematical morphology is insufficient and the output image cannot be correctly separated. Furthermore box and rocher classes are virtually indistinguishable.

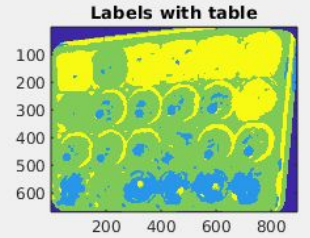
Result not acceptable











CONTRIBUTIONS

Gianluca Giudice 60%

- Image binarization
- Feature analysis
- Classifier
- Error plot

Matteo Formenti 40%

- Vertices calculation
- ROI isolation
- Slides