Sorrentino Alessandro, 815999
Ratti Burt, 816243
Zuccarella Stefano, 816482

Products classifier

Objective

The main objective is to recognize a series of products in a given image and print the receipt.

Case of application: automated checkout registers

Preconditions

- The images were taken from above at a constant distance $d \pm \varepsilon$, in order to have more or less the same size for the objects.
- The lighting is slightly not homogeneous because of flash (radial illumination from the center).
- Objects can slightly touch other objects and they can be rotated in different ways.
- Objects should have as less shadows as possible.
- The background color must be homogeneous and there must not be other things besides that.
- The images were taken with the photocamera Canon EOS 700D, lens 18-55 mm.

Objects to recognize /1

Algida

Aranciata

Cioccolato

Cocacola

Ghiaccioli

Integrale

Limone

Mandarino

Mela

Objects to recognize / 2

Milka

Pasta

Viviverde

Yomo

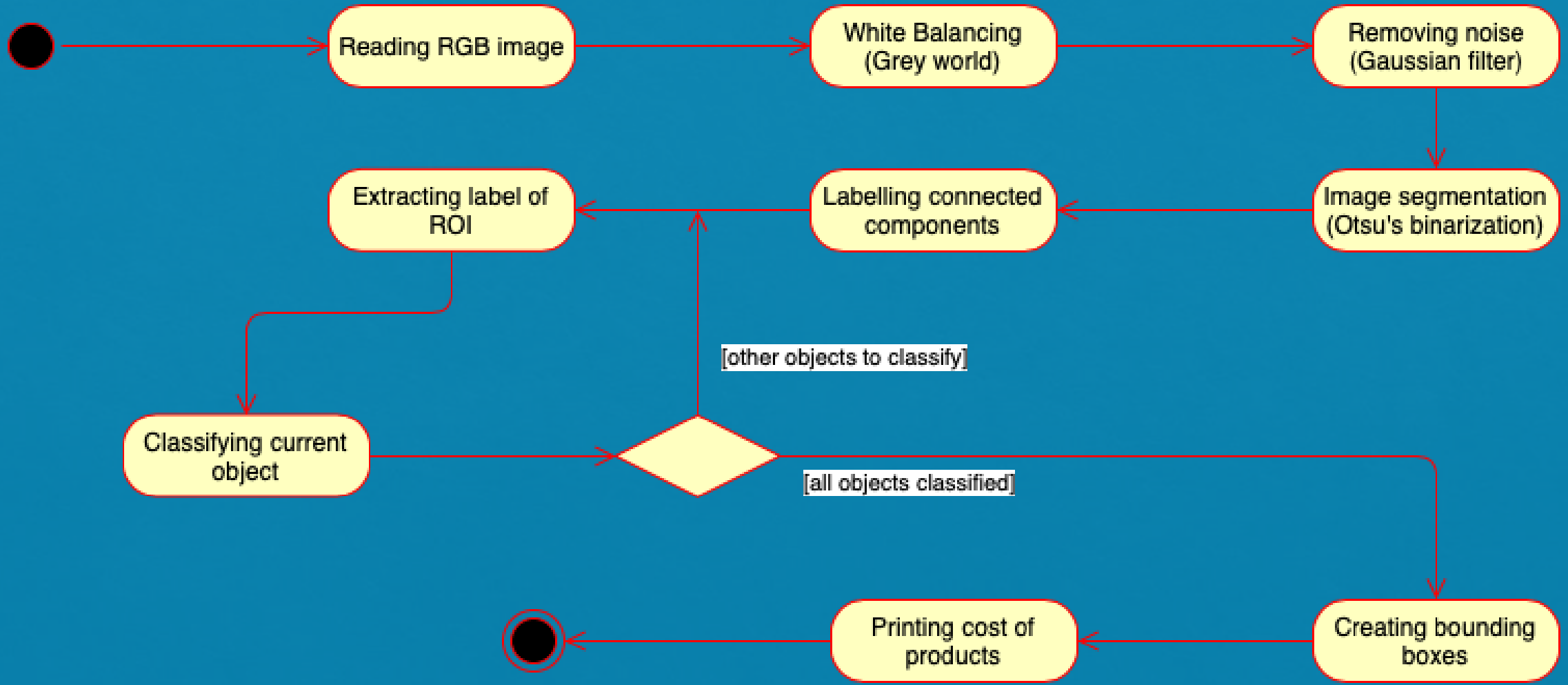
Sprite

The

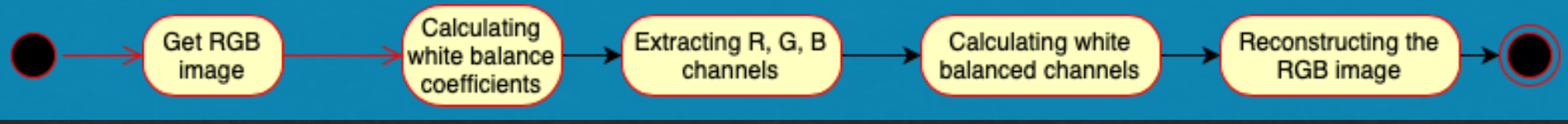
Passata

Rummo

Generic pipeline



White Balancing

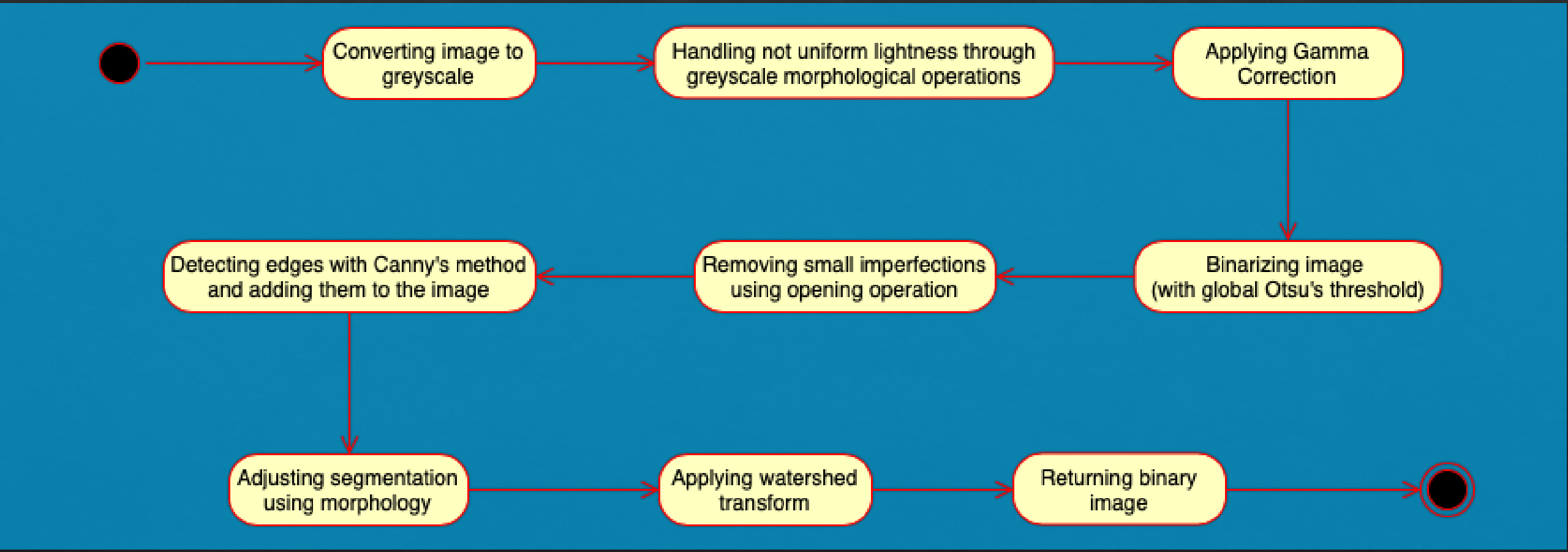


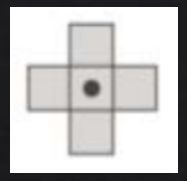
White balancing example using *"Gray world" algorithm*

- Rx = Kx * X, Kx = GrayX/Xaverage, GrayX = 128 (or 0.5) (X € {R, G, B})

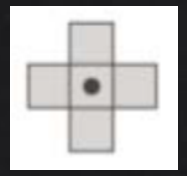
(Images have been taken using flash in order to reduce the presence of shadows and to uniform the color. This image has been taken without it to show the management for photos without flash)

Segmentation pipeline





*Structuring element for **morphology** is a disk of size 3x3*



*Structuring element for **background removal** is a disk of size 60x60*

Other parameters:

Gamma correction: gamma = 1, low_in & high_in = 0,

Opening for imperfections: 1000(px) (Area)

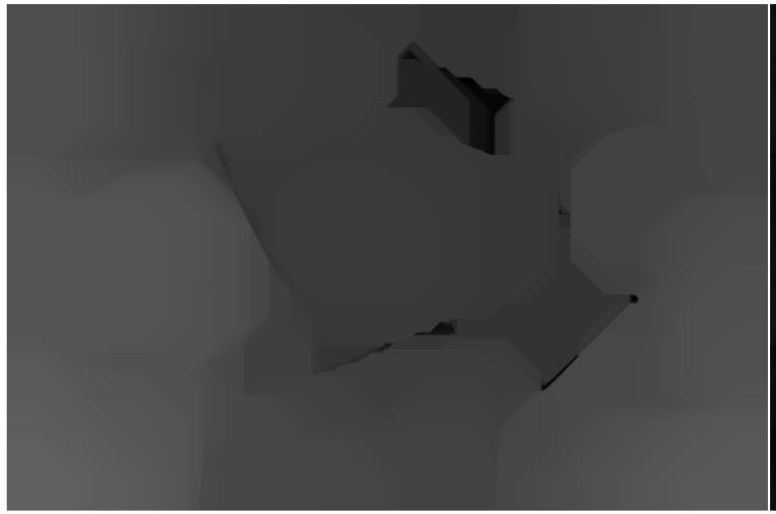
Canny edge magnitude: 0.1

Segmentation

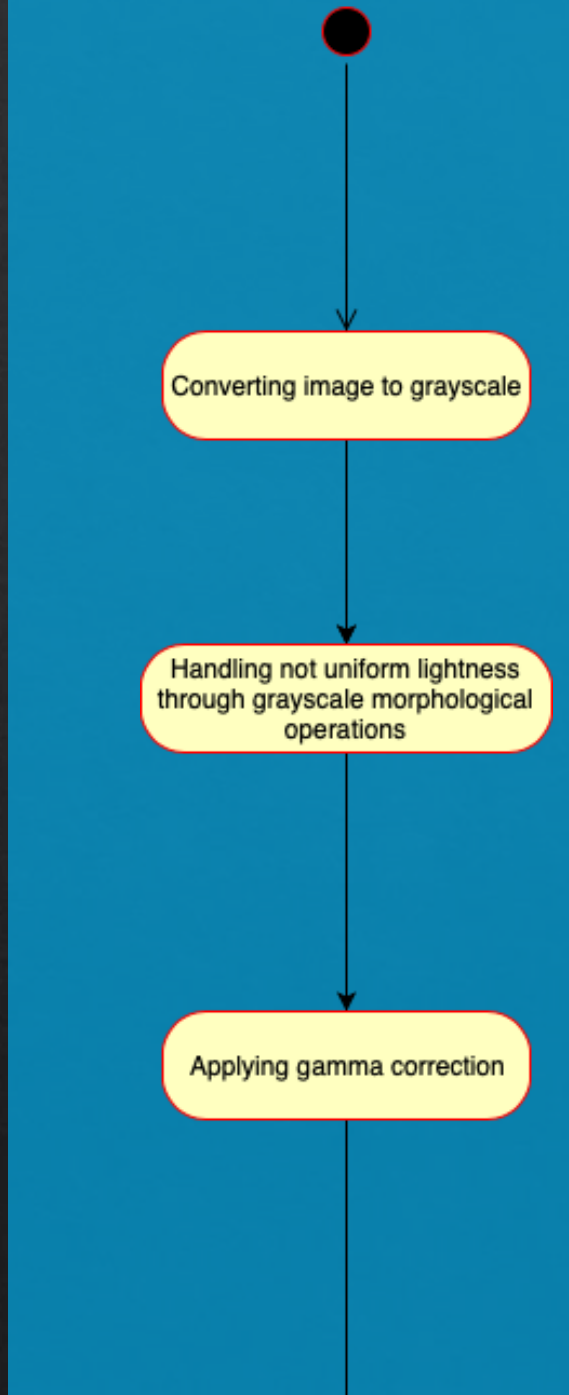
Greyscale image

Complementary image

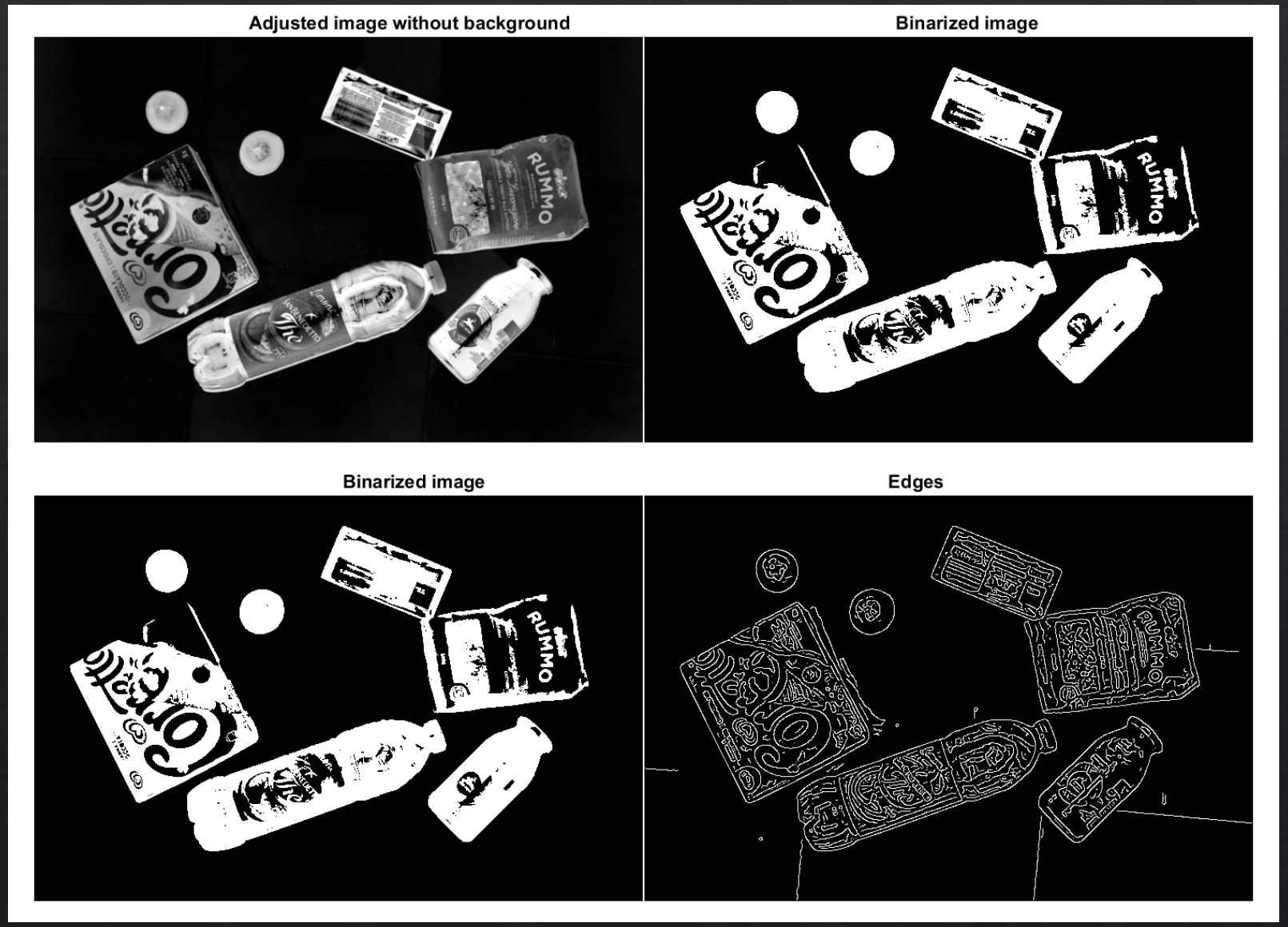
Background



Adjusted image without background



Segmentation



Binarizing image (With global Otsu's threshold)

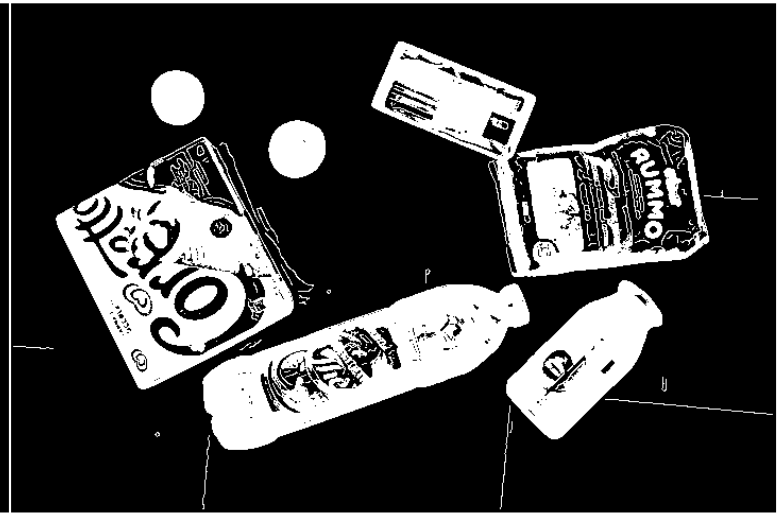
Removing small imperfections using opening operation

Detecting edges with Canny's method

Segmentation

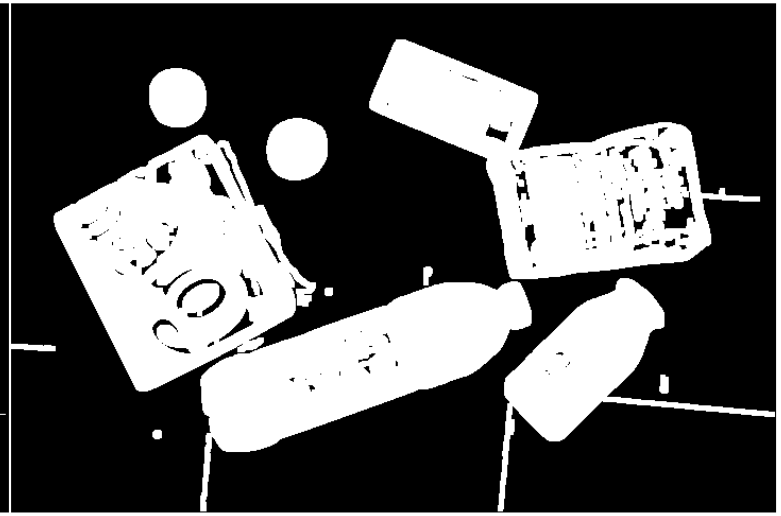
Binarized image

BW | Edges



BW | Edges

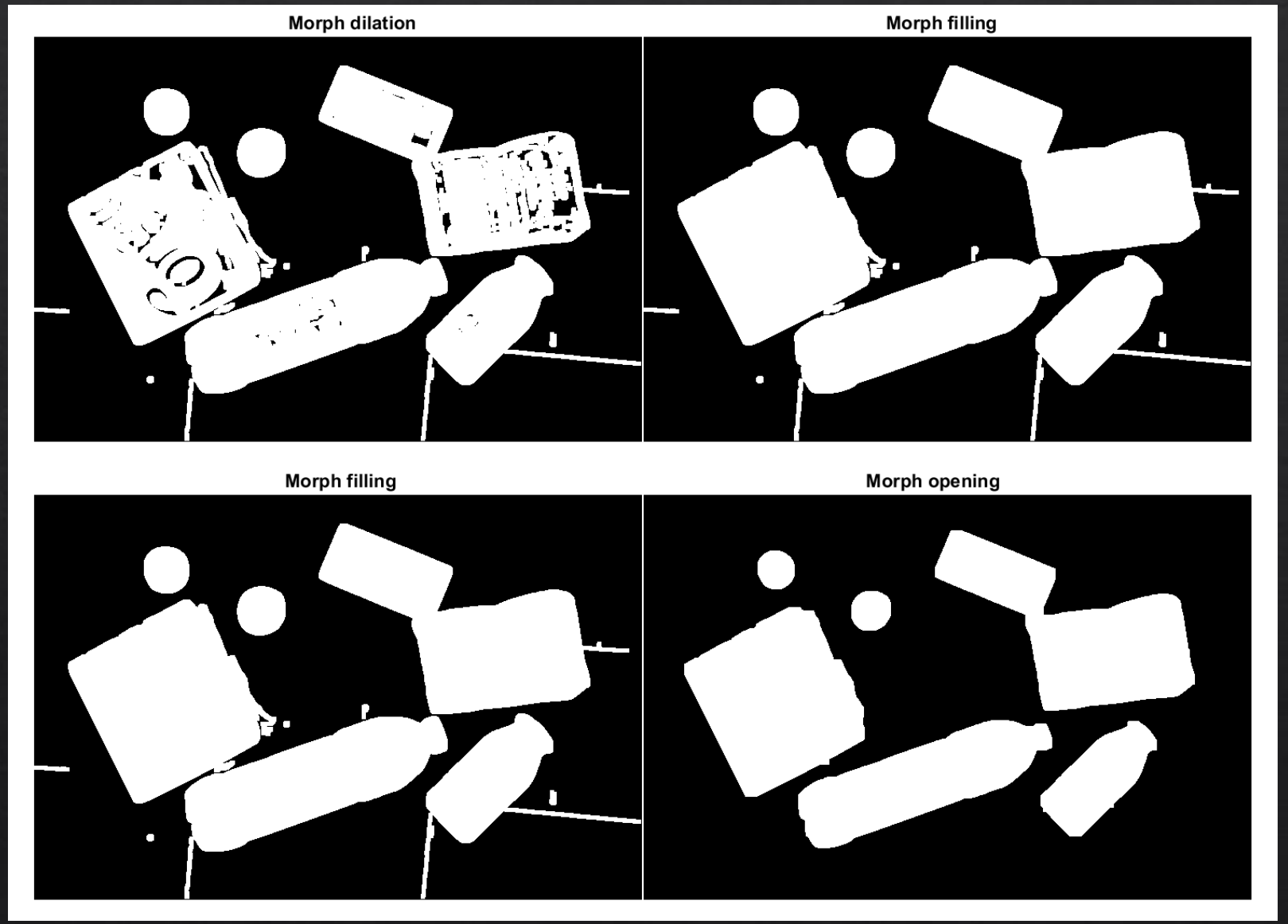
Morph dilation



logic OR between binarized image and Canny edges

Morph dilation to close object borders

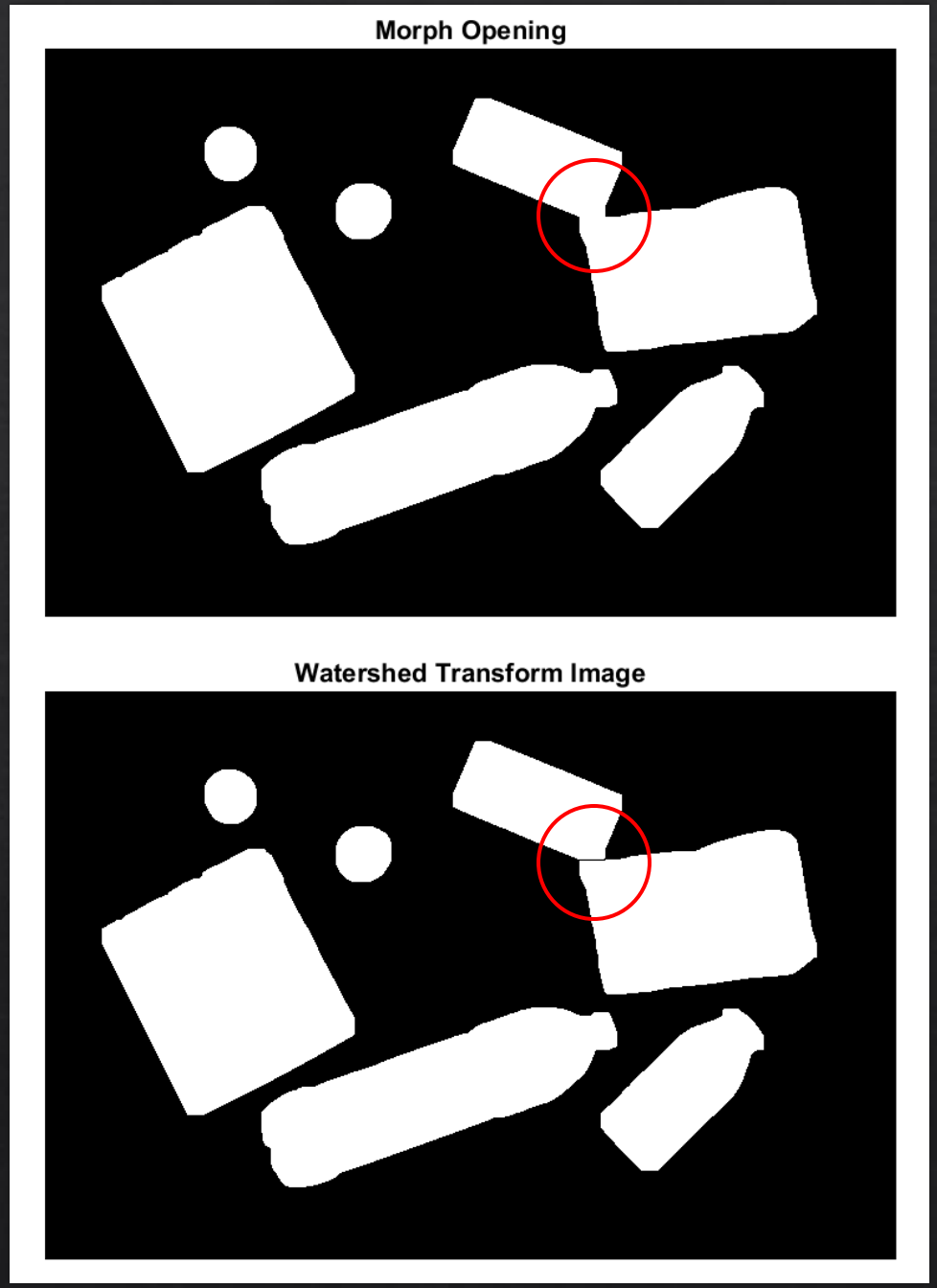
Segmentation

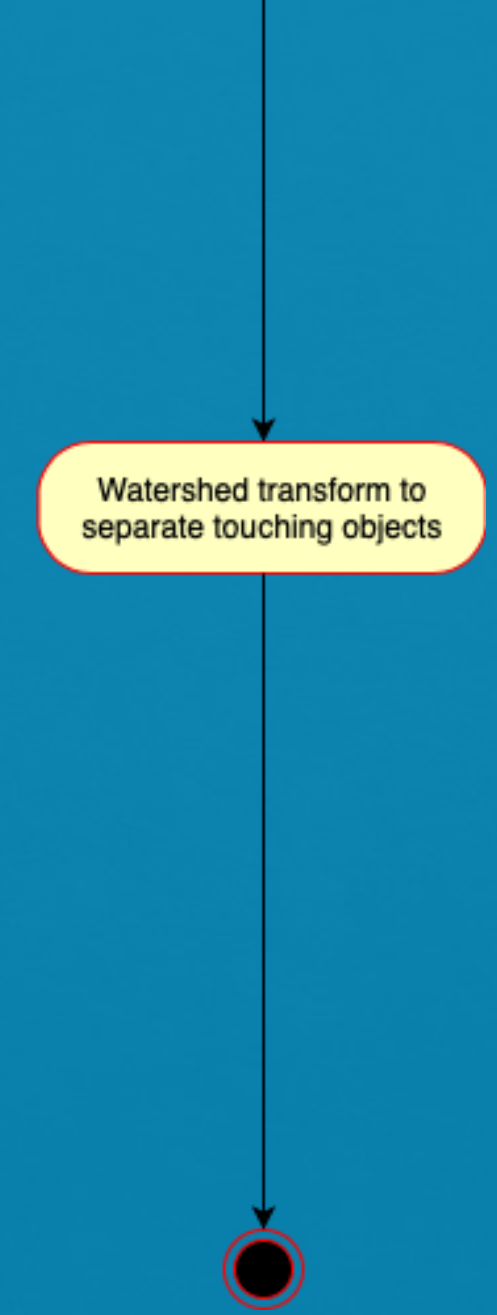


Hole filling

Opening to delete spurious regions

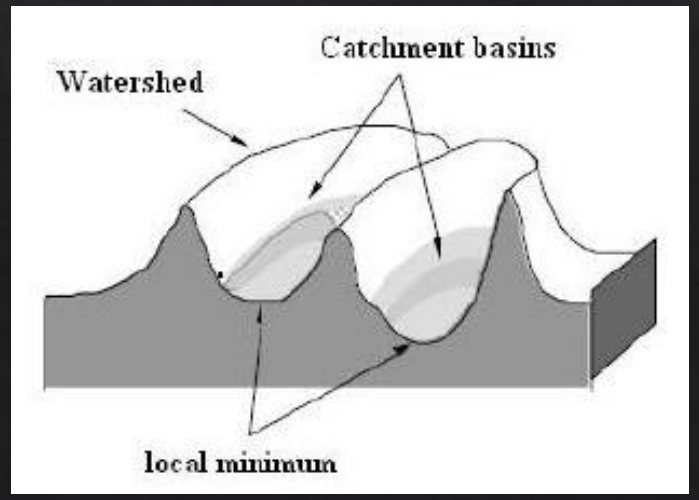
Segmentation

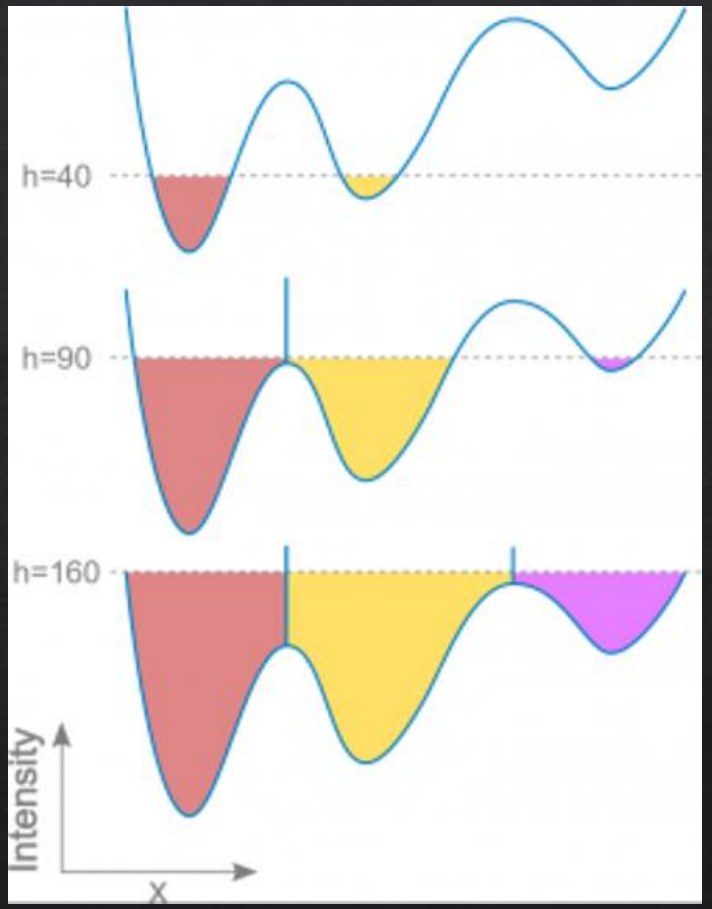




Watershed Transform

- Input: Greyscale Image
- The image can be seen as a topological surface, where pixel values correspond to heights.
- The main idea is to flood from below (local minimum) the surface with water at a uniform rate. When two flooded regions from separate catchment basins collide, a watershed ridge is created to avoid the merger (edge):
- It is especially useful for segmenting touching objects.
- Output: Label Matrix -> The elements labeled 0 belong to watershed ridges, while the elements labeled with other nonnegative integers belong to the regions identified.

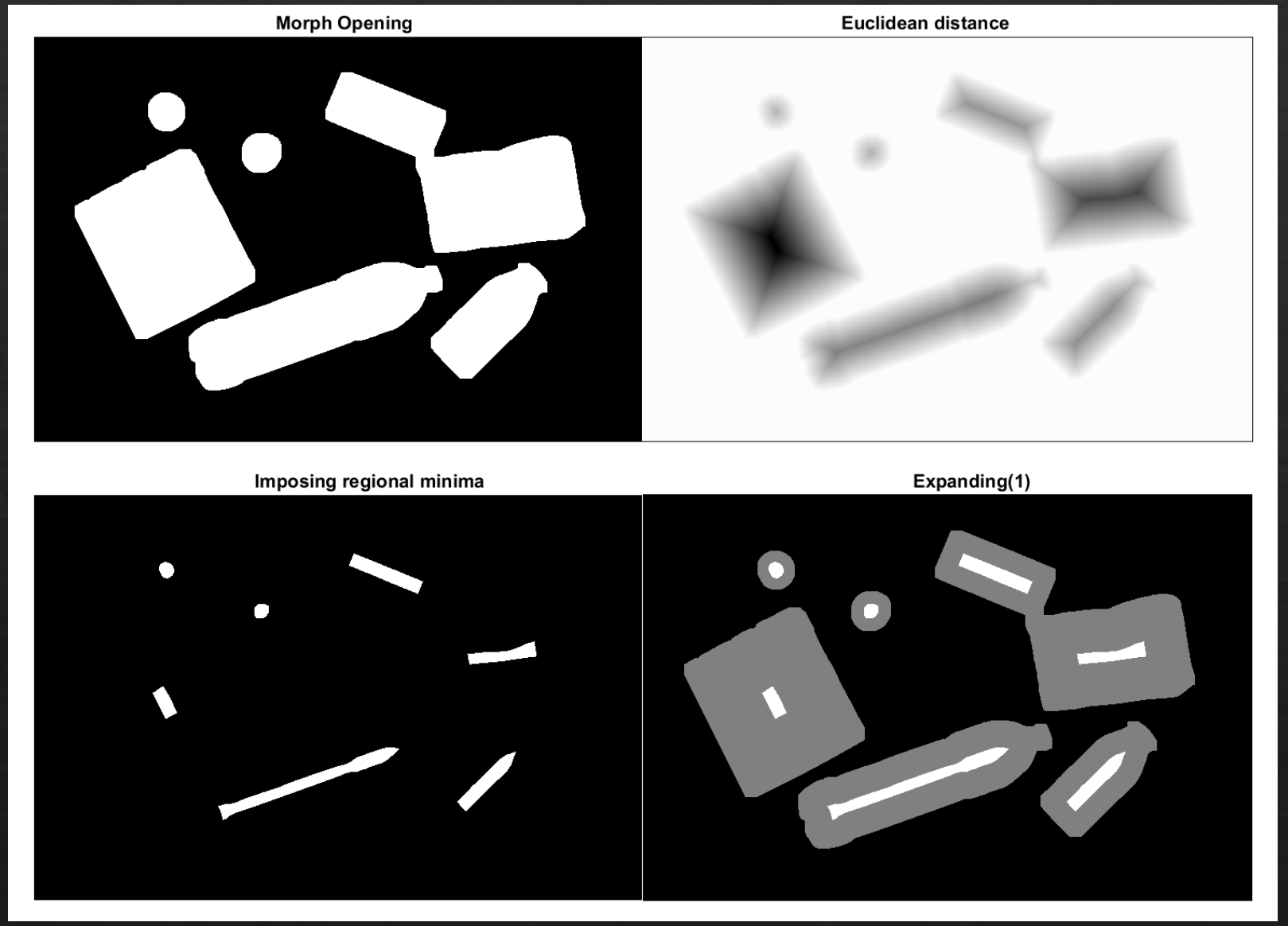


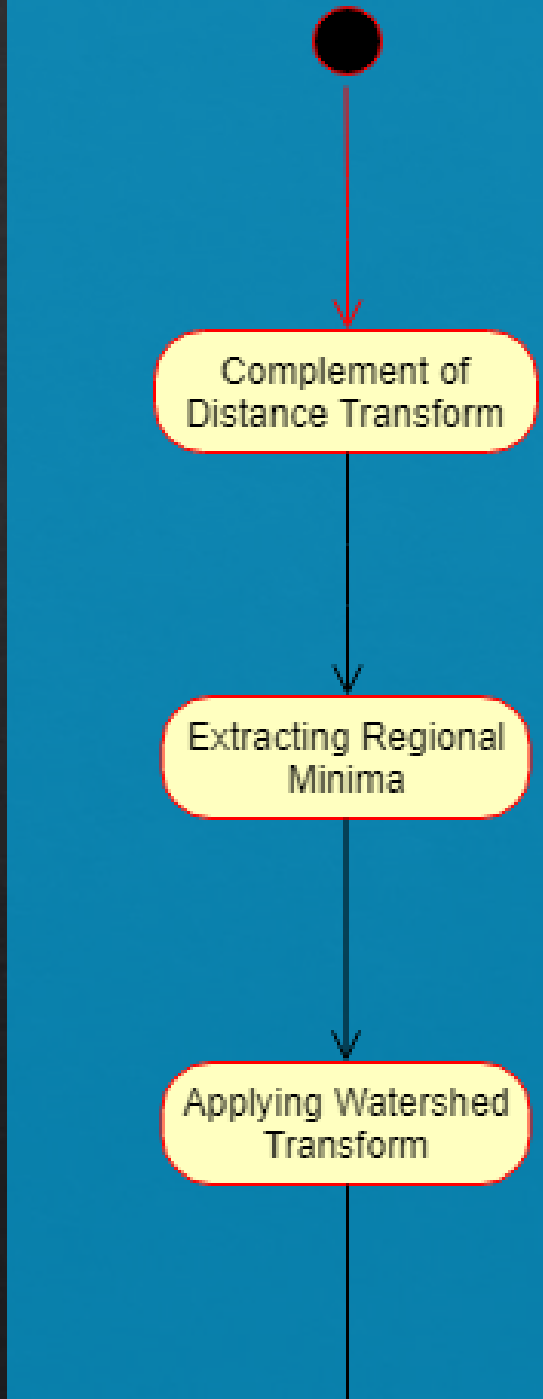


Parameters used:

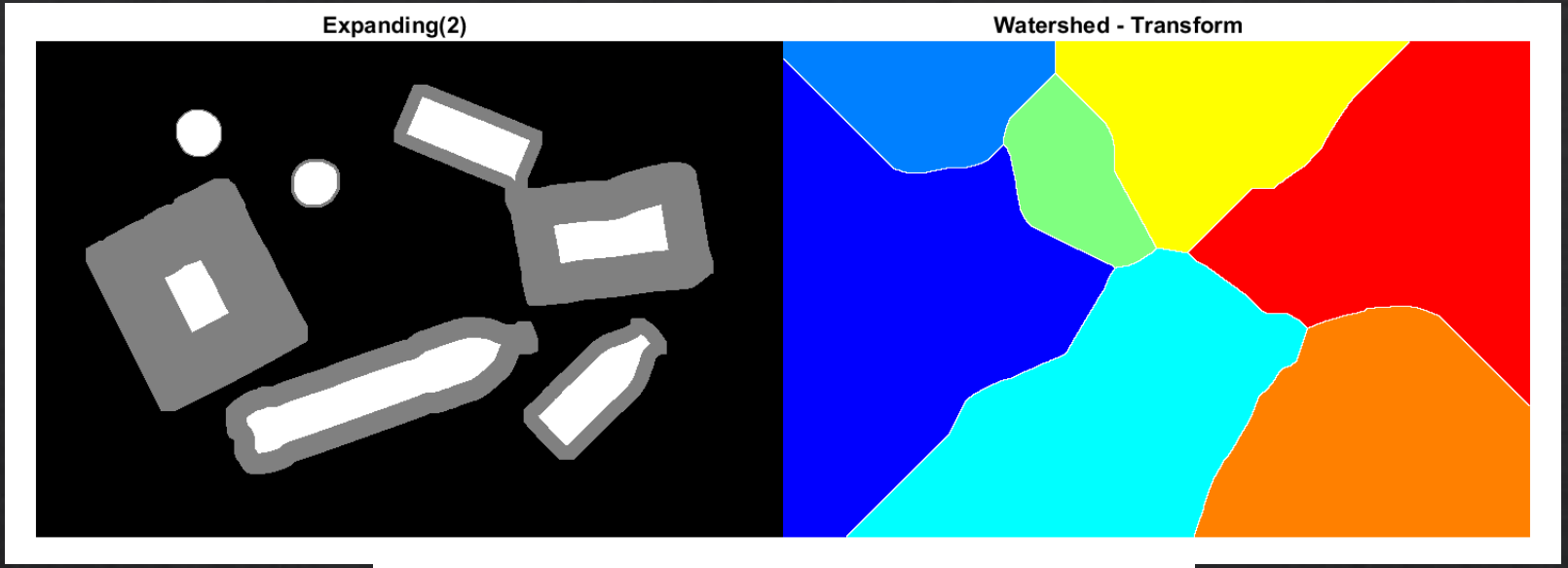
ExtendedMinima: 8 (H-minima transform, min is not 0 anymore but starting from height H)

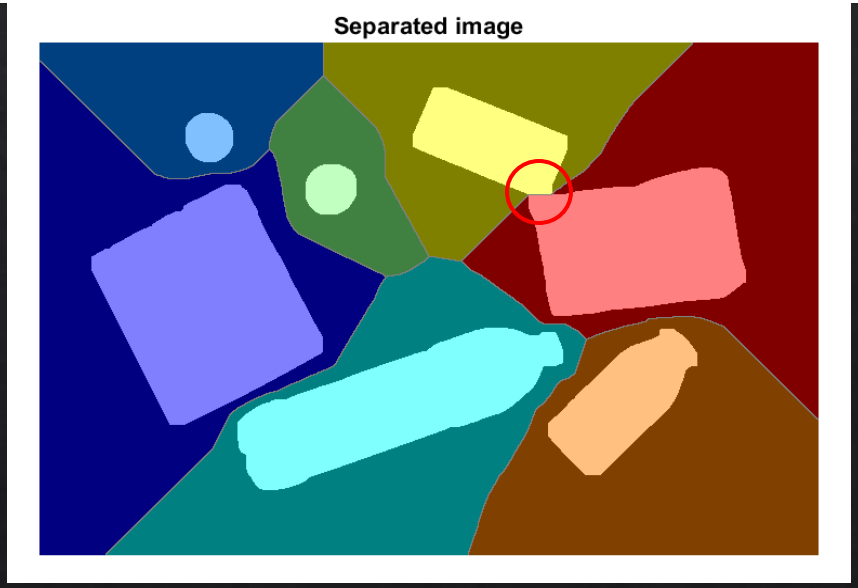
Watershed Algorithm

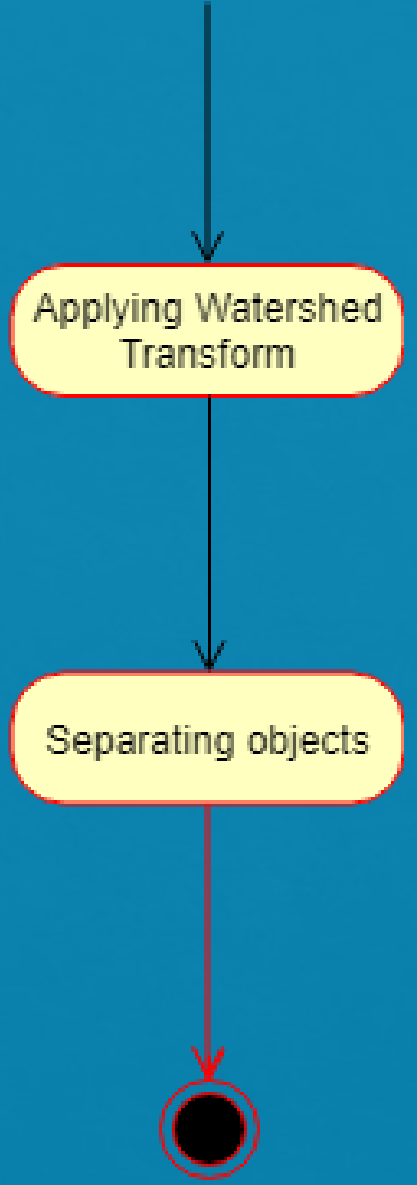




Watershed Algorithm



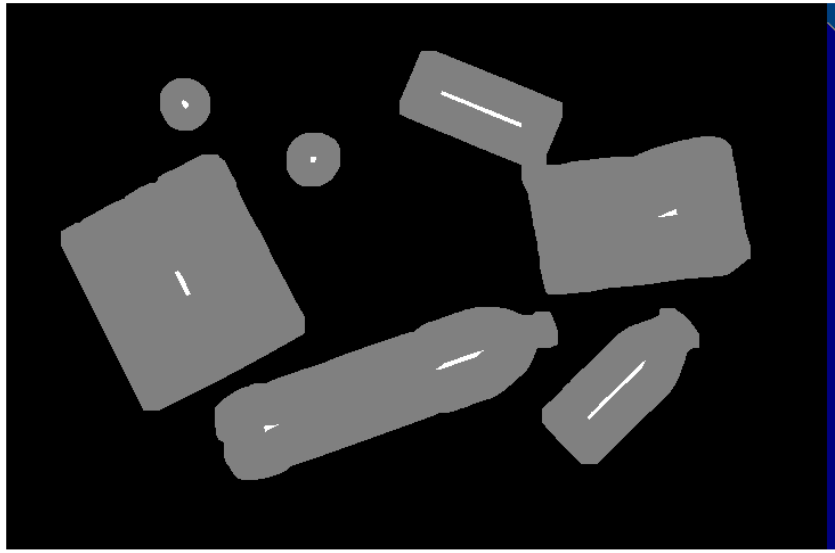




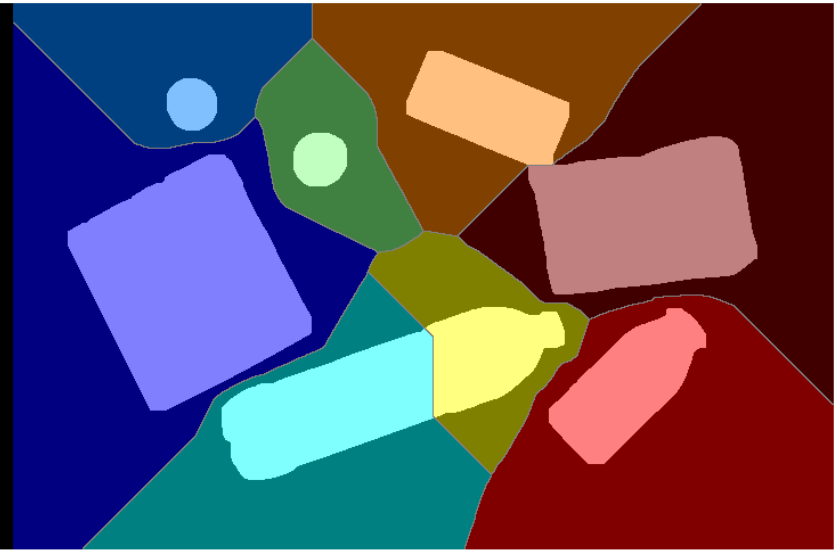
Watershed transform

Parameter for extended minima

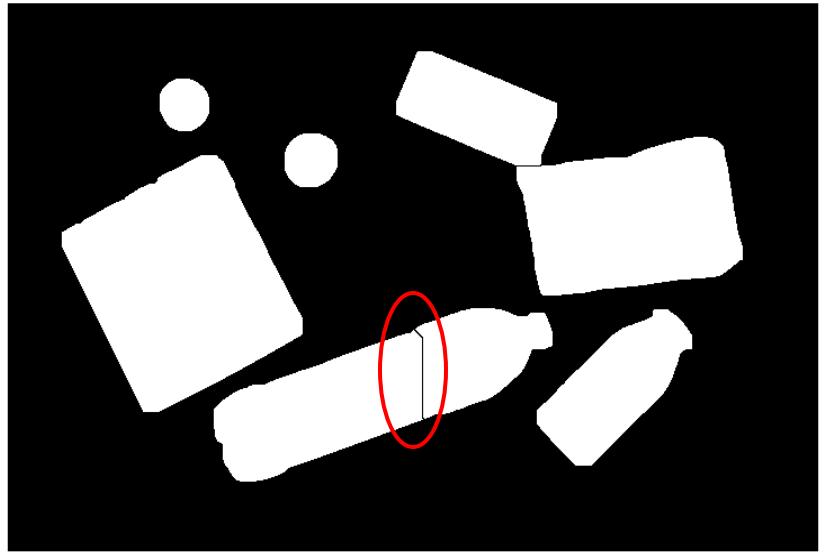
Bad Regional Minima Imposition



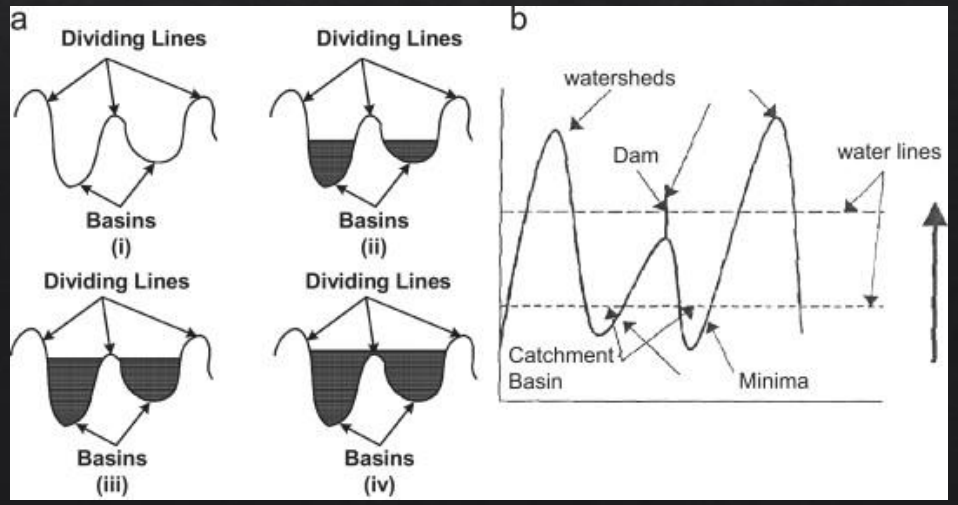
Separated Image



Segmented Image



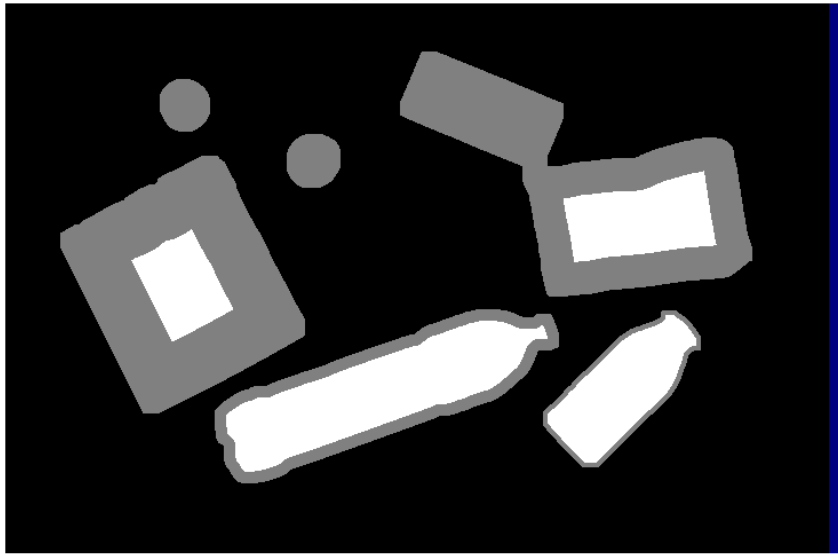
We can see that if the H-extended minima is too low, the object is cut down in two parts. So when performing the watershed a single objects is now divided in two parts since there are two local minima inside one object



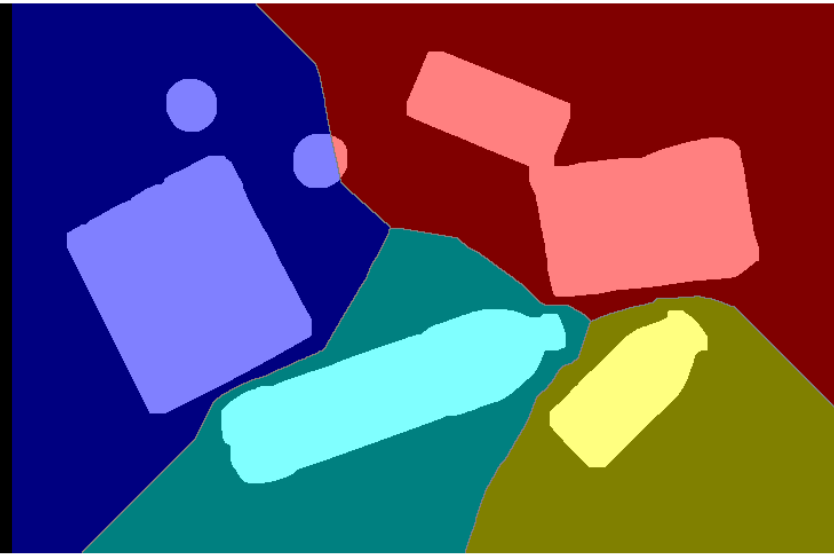
Watershed transform

Parameter for extended minima

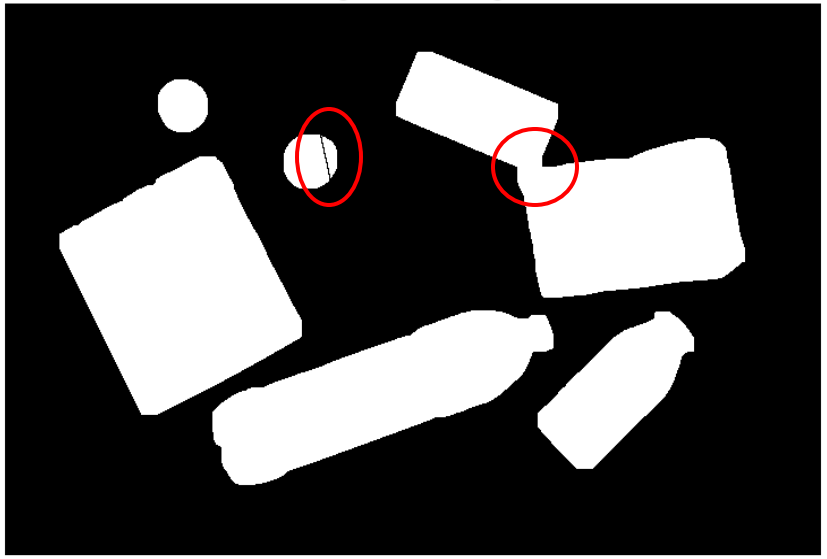
Bad Regional Minima Imposition



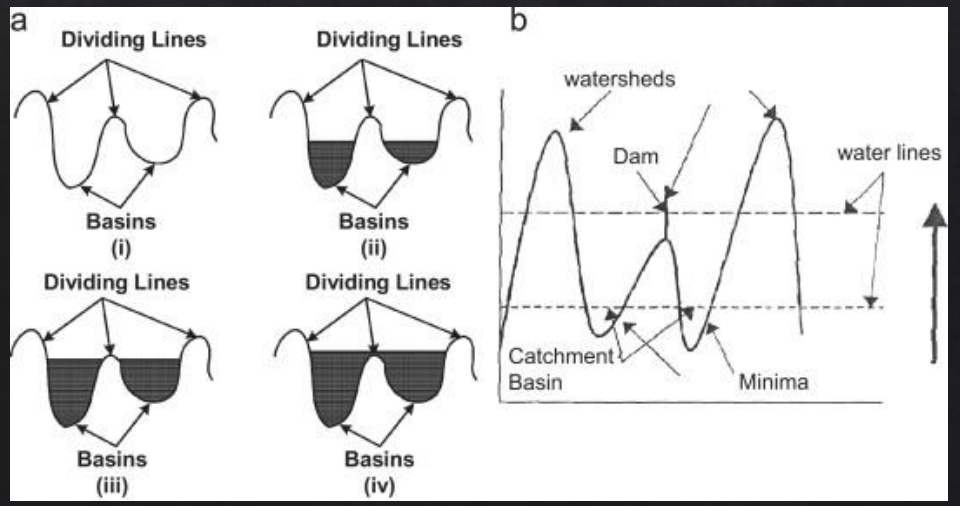
Separated Image



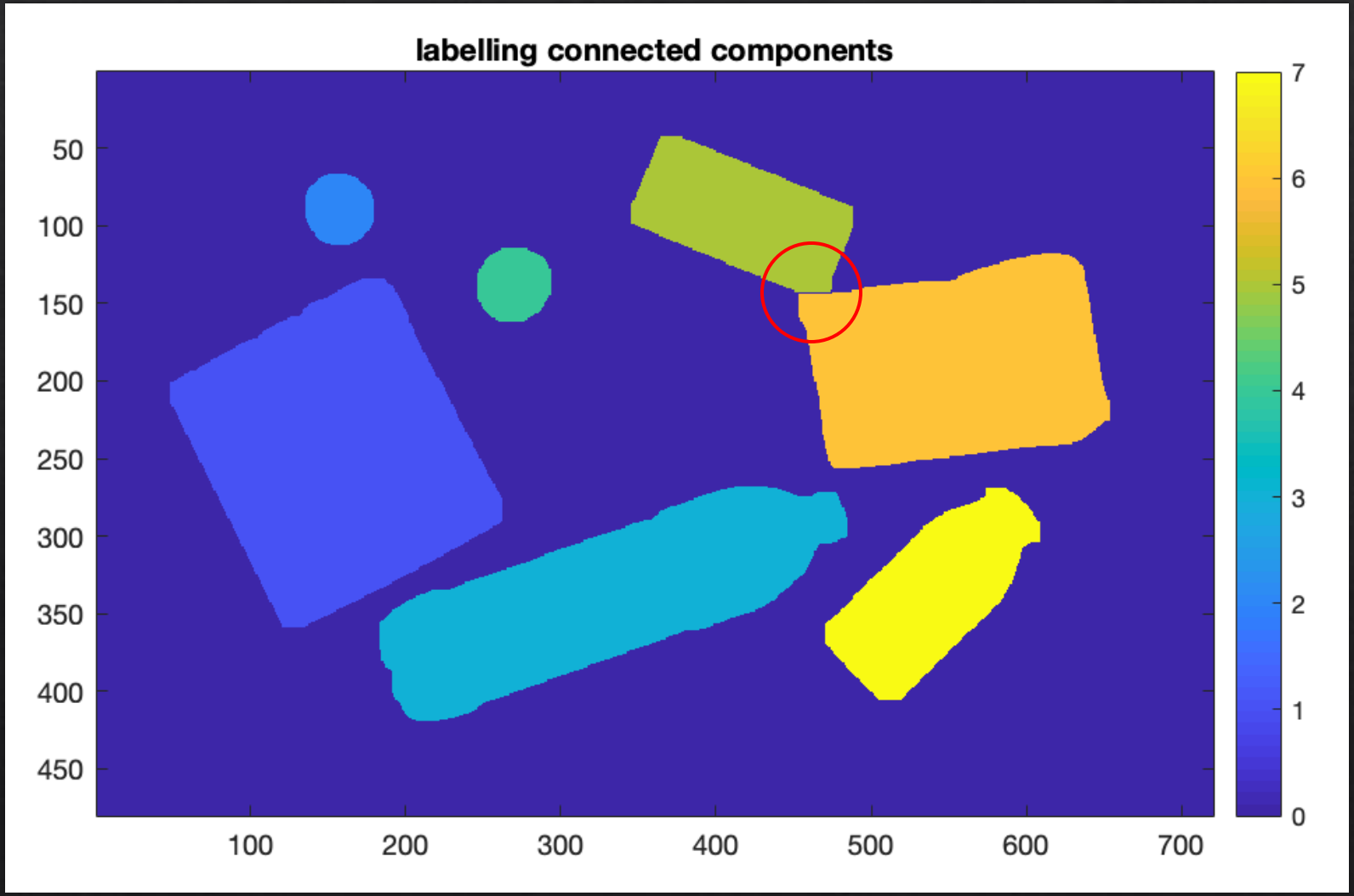
Segmented Image



If the H-extended minima parameter is too high, the minima of that level is too high so no minima value is found. That leads to very bad results



Labelling connected components



Thanks to watershed transform, touching objects are considered separated during labelling since an artificial disconnection was made

Extraction of ROI

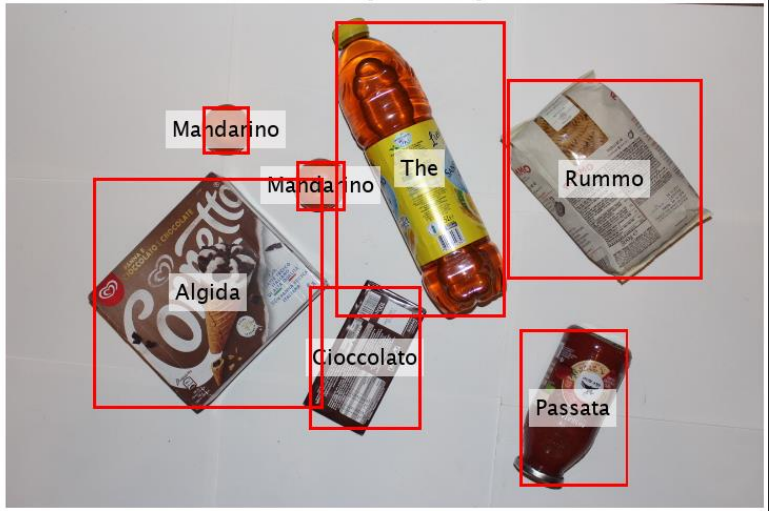
The object is obtained by applying a mask containing the i-label to the image. Then, the ROI is detected by cropping the region of the object and passed to the classifier.

Bounding Boxes + Receipt

Original image

Classified image

Bounding box image



Receipt

```
  Product Classifier
      SCONTRINO
Algida          3.50€
Mandarino       0.20€
Mandarino       0.20€
Cioccolato      1.20€
The             2.50€
Passata         1.20€
TOTALE          8.80€
```

A bounding box is drawn for each classified product. At the end, the receipt is written, not including unknown objects.

Classification
(Classificator)

We used a *KNN classificator* with the following parameters:

Number of neighbors: 3

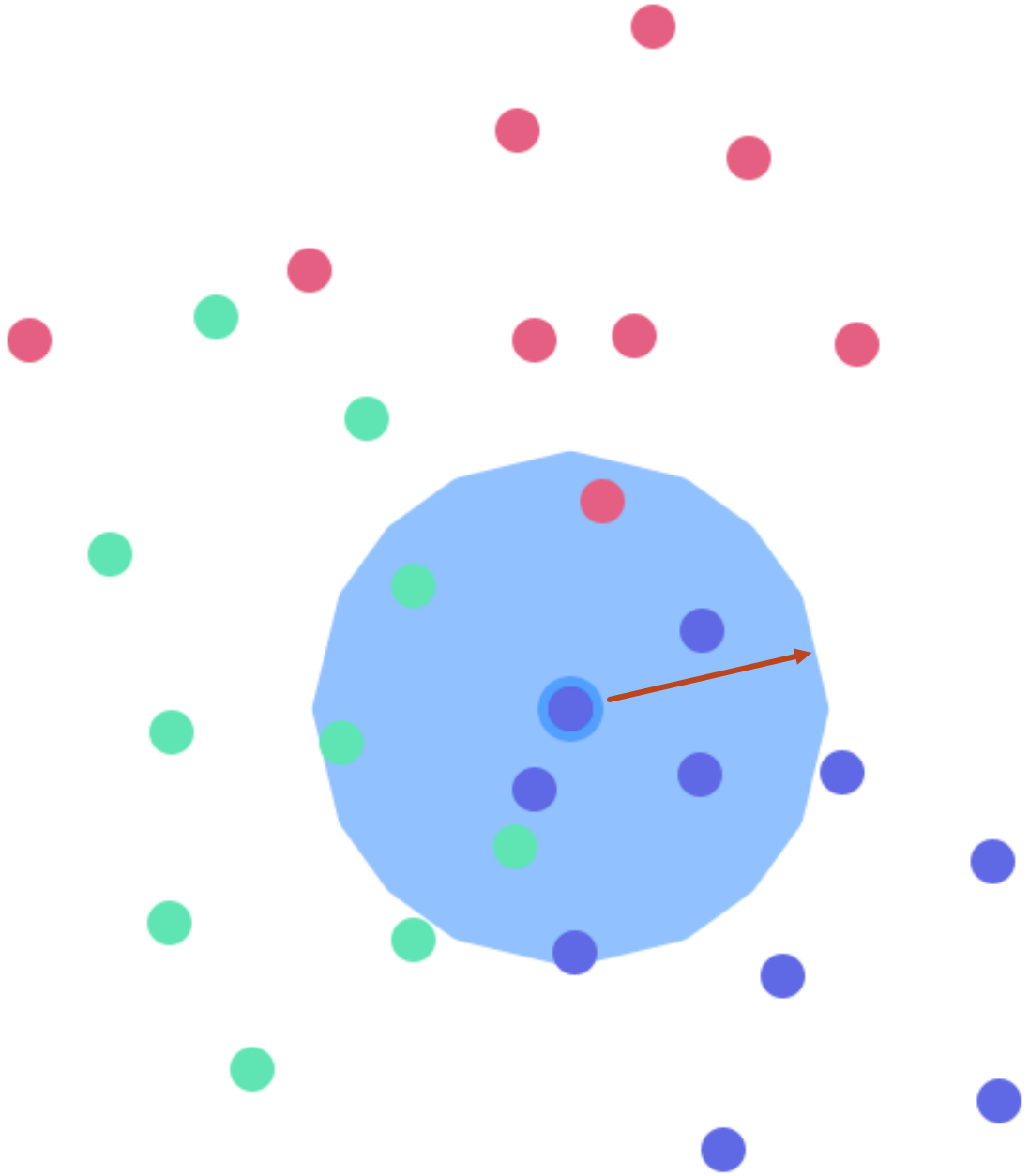
Distance: 'Euclidean'

Weight on distance: squared inverse

$$\frac{1}{dist^2}$$

Rejection threshold on probability:

T < 0.65



Classification

(Feature vector)

Which feature discriminates the most the objects we have in our dataset?

Shape

- Hu invariant moments
- Compactness

Color

- Mean RGB color

Texture

- CEDD

Feature vector

$$x = [\ \overline{hu} \quad \overline{comp} \quad \overline{color} \quad \overline{cedd}\ ]$$

Classification

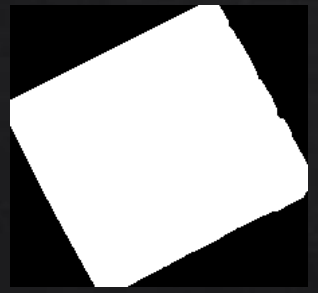
(Compactness, hu & color)

Compactness

Describes type of shape with a numerical value.

$$C = \frac{4\pi A}{2P^2}$$

Example



$$C = 0.84673238$$

Color

Mean color of each channel separately (RGB)

Color is invariant for scale, rotation and translation

Example

$$meanR = 0.224...$$

$$meanG = 0.143...$$

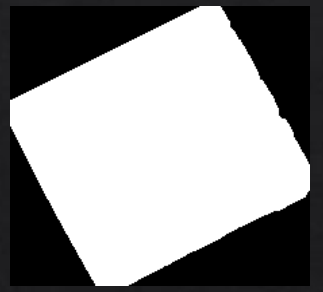
$$meanB = 0.366...$$

Hu moments

Describes shape using central moments

Hu moments are invariant for scale, rotation and translation

Example



$$h_1 = 0.1669...$$

$$h_2 = 3.7466... \cdot 10^{-4}$$

$$h_3 = 6.9359... \cdot 10^{-7}$$

$$h_4 = 5.6463... \cdot 10^{-7}$$

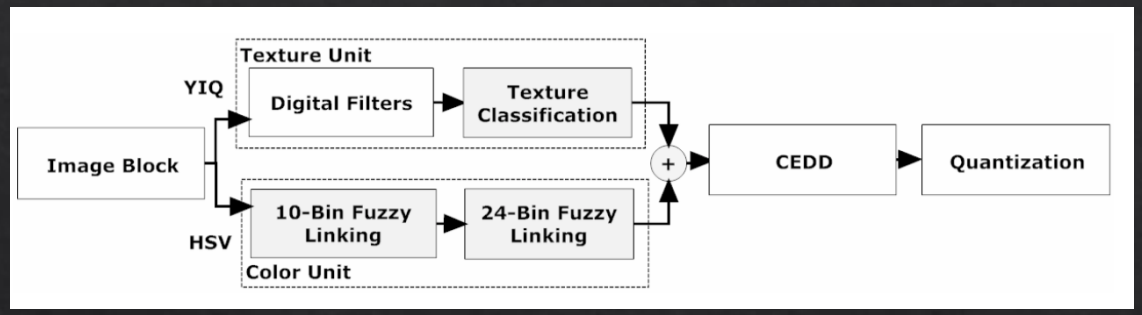
$$...$$

Classification
(CEDD)

Calculates the Color and the Edge Directivity Descriptor (144 values)

Example

$$CEDD\_1 = 1.6552$$
$$CEDD\_2 = 0.5959$$
$$CEDD\_3 = 3.7738$$
$$CEDD\_4 = -0.4634$$
$$\dots$$



Color unit

- Image divided in blocks
- RGB > HSV
- HSV > 10 bin histogram

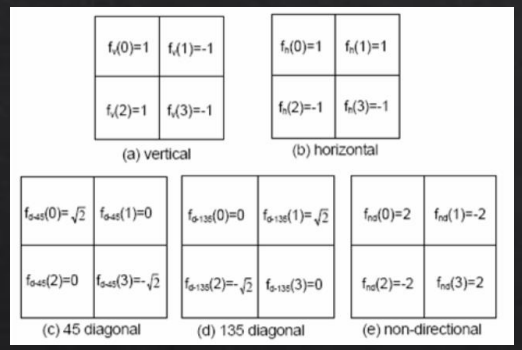
Black, Gray, White, Red, Orange, Yellow , Green, Cyan, Blue, Magenta

$$I - (I \otimes CLF) = vertical\ edges$$

- Each channel of HSV divided in fuzzy reagions (using edge info)
- Multi Participate Algorithm > defuzzyfication
- Second system > color divided in 3 hues > 24 bin histogram

Dark Color – Color (10 bins from before) - Light color

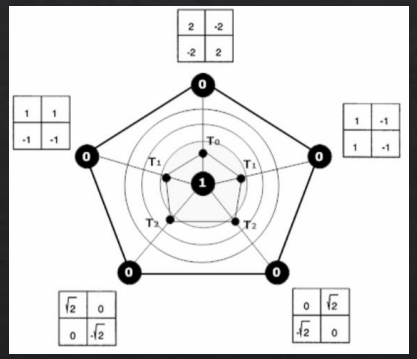
- Defuzzification as before (each bin has number of values according to defuzzyfication algorithm)



Texture unit

Each block is divided In 4 sub blocks

Calculating edge on each sub blocks



(vertical, horizontal, 45° diagonal, 135° diagonal, on-directional edge)

Calculating magnitude for each block for each type of edge + normalization (mk /max)

6 area histogram > non-edge + edge (6 bins total)

For each bins count how many edges of that type there are with thresholds value (**texture classification**)

Units combination

Texture unit = 6 areas

Color unit = 24 areas

Total areas= 6 x 24 = 144 areas
6 areas of texture unit for each area of color unit

Normalization {0, 1}

Classification

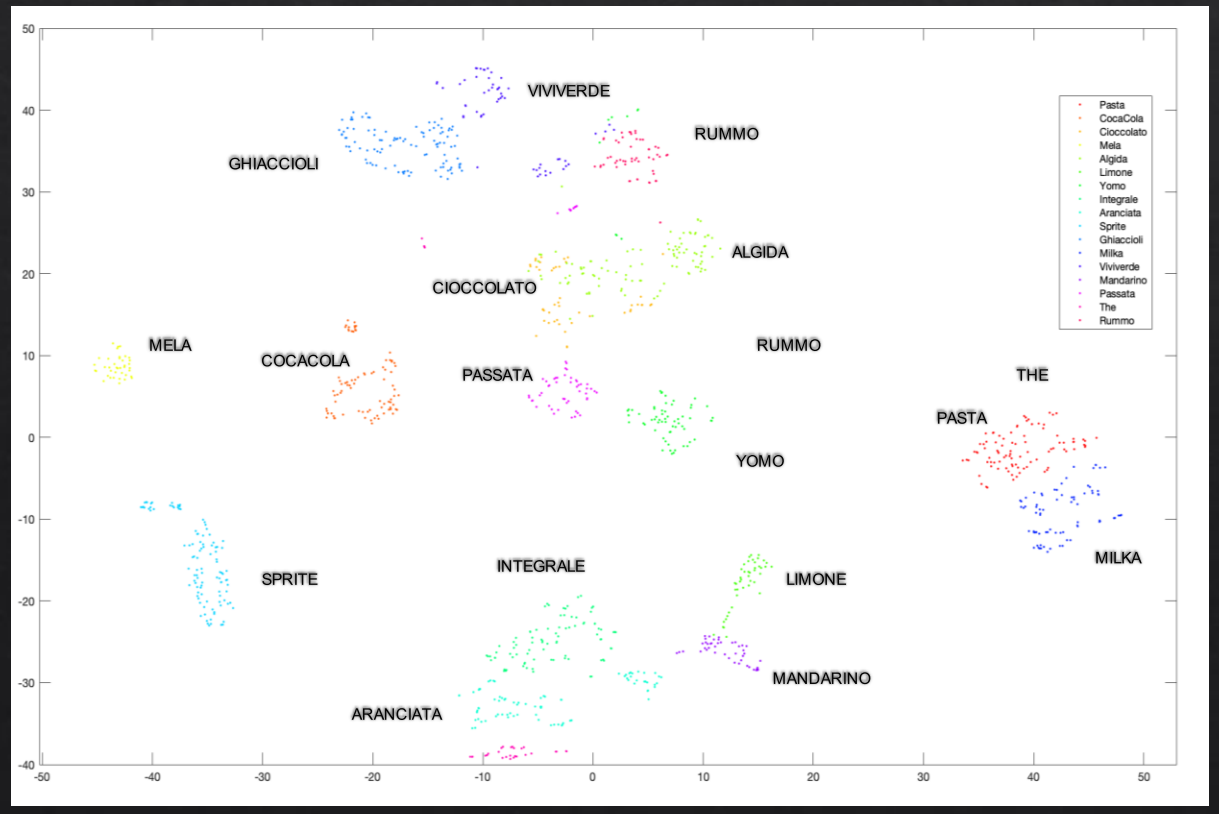
(Feature vector normalization)

Normalizing each group of feature inside feature vector

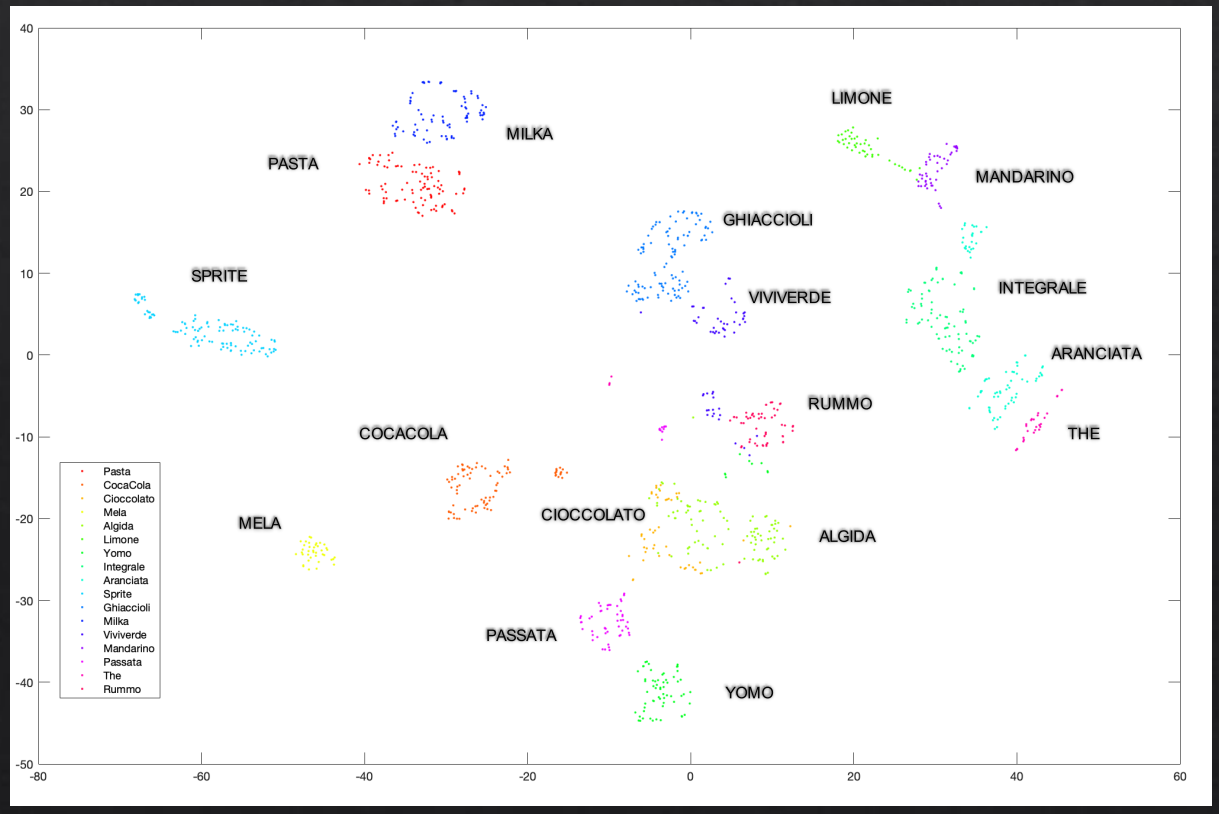
$$fnormalized = \frac{f - \mu(f)}{\sigma(f)} \quad \forall f_i \in featurevector$$

Feature space using t-SNE

Feature vector



Normalized feature vector



Correct images

(ERROR)

Wrong images

The classification errors are probably caused by:

Tomato jar's color has the same brownish hue of the chocolate bar. That's probably a consequence of post-processing of the camera (in other test photos tomato jar is having a nice red hue)

The chocolate is set to unkown because of rejection threshold. The classifier is pretty insecure since the standing algida has the same mean color, texture and compactness of the chocolate bar

The classification errors are probably caused by:

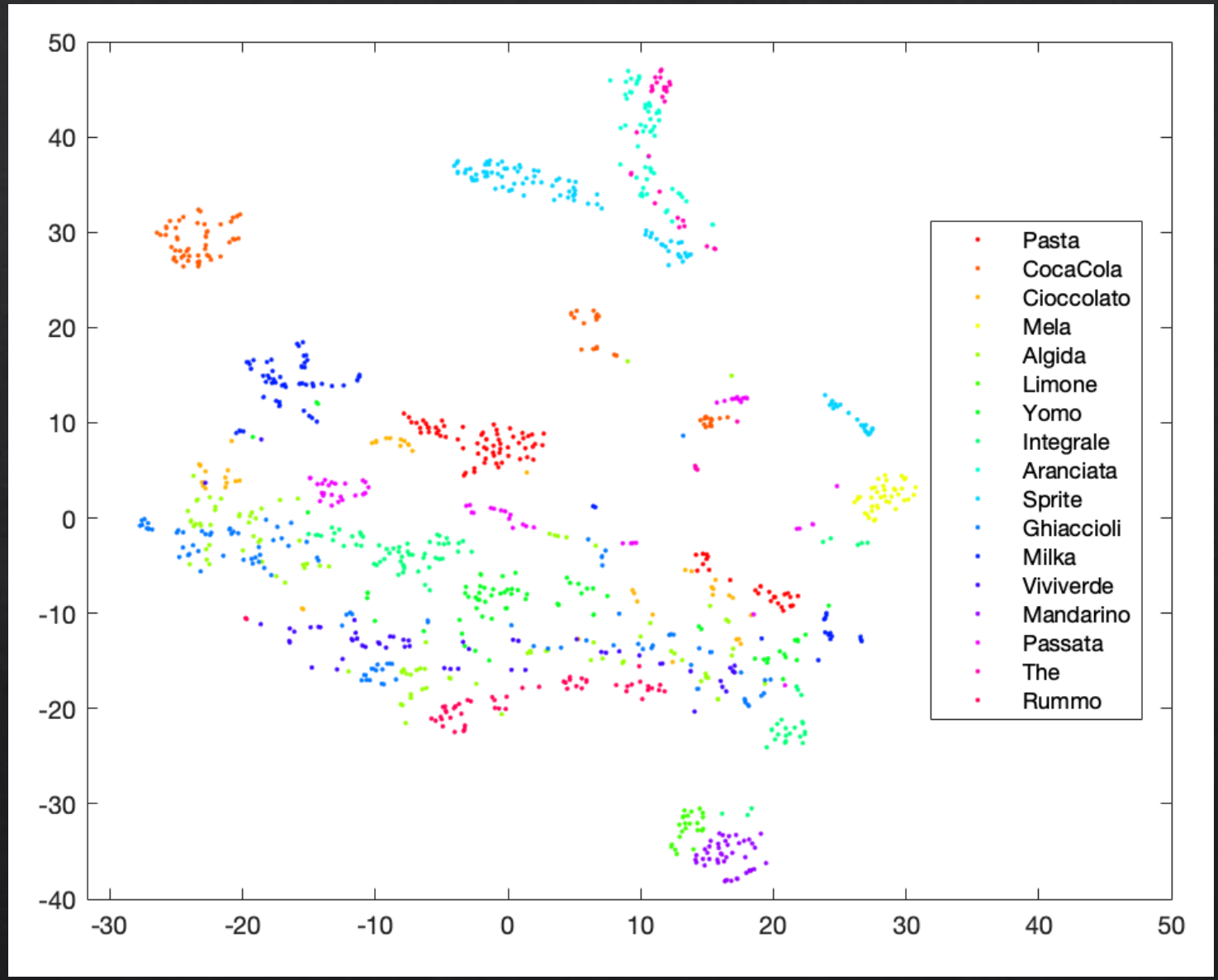
The problem with hue yogurt and novi chocolate bar is probably the same as described before.

Furthermore, there's a dominant light orange light which could have been handled with a chroma adaptation

Performance

RELEASE 1:

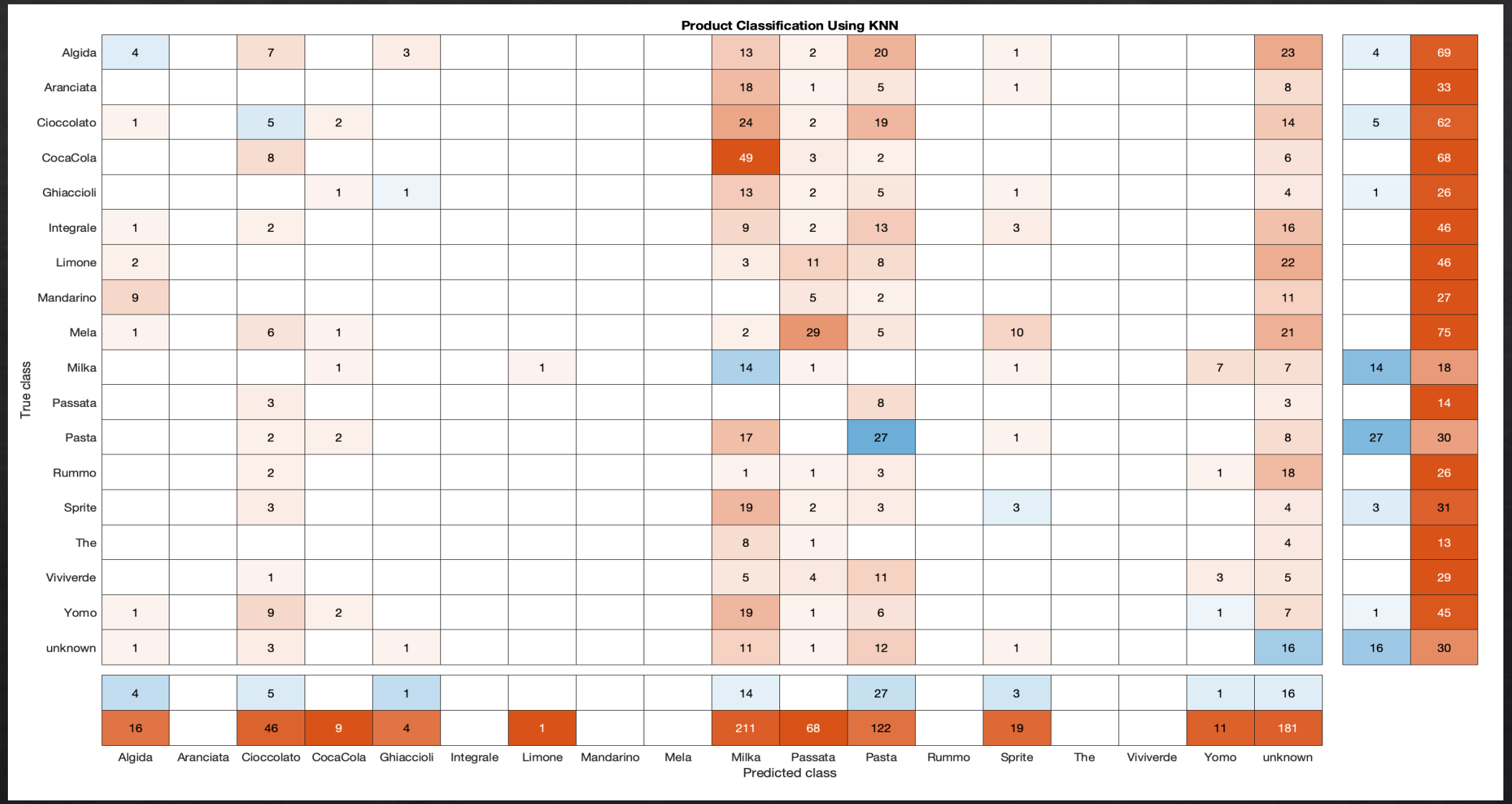
- NO touching objects or overlapping objects
- NO standing objects or slightly prospective
- White balancing (Gray world)
- Feature [hu, LBP, meanColor]



Performance

RELEASE 1:

- NO touching objects or overlapping objects
- NO standing objects or slightly prospective
- White balancing (Gray world)
- Feature [hu, lbp, meanColor] (not so discriminant)



Performance

RELEASE 1:

- NO touching objects or overlapping objects
- NO standing objects or slightly prospective
- White balancing (Gray world)
- Feature [hu, LBP, meanColor]

As you can see, features like hu, lbp and mean color are not that discriminant as we initially thought.

There are a lot of objects that have similair mean color.

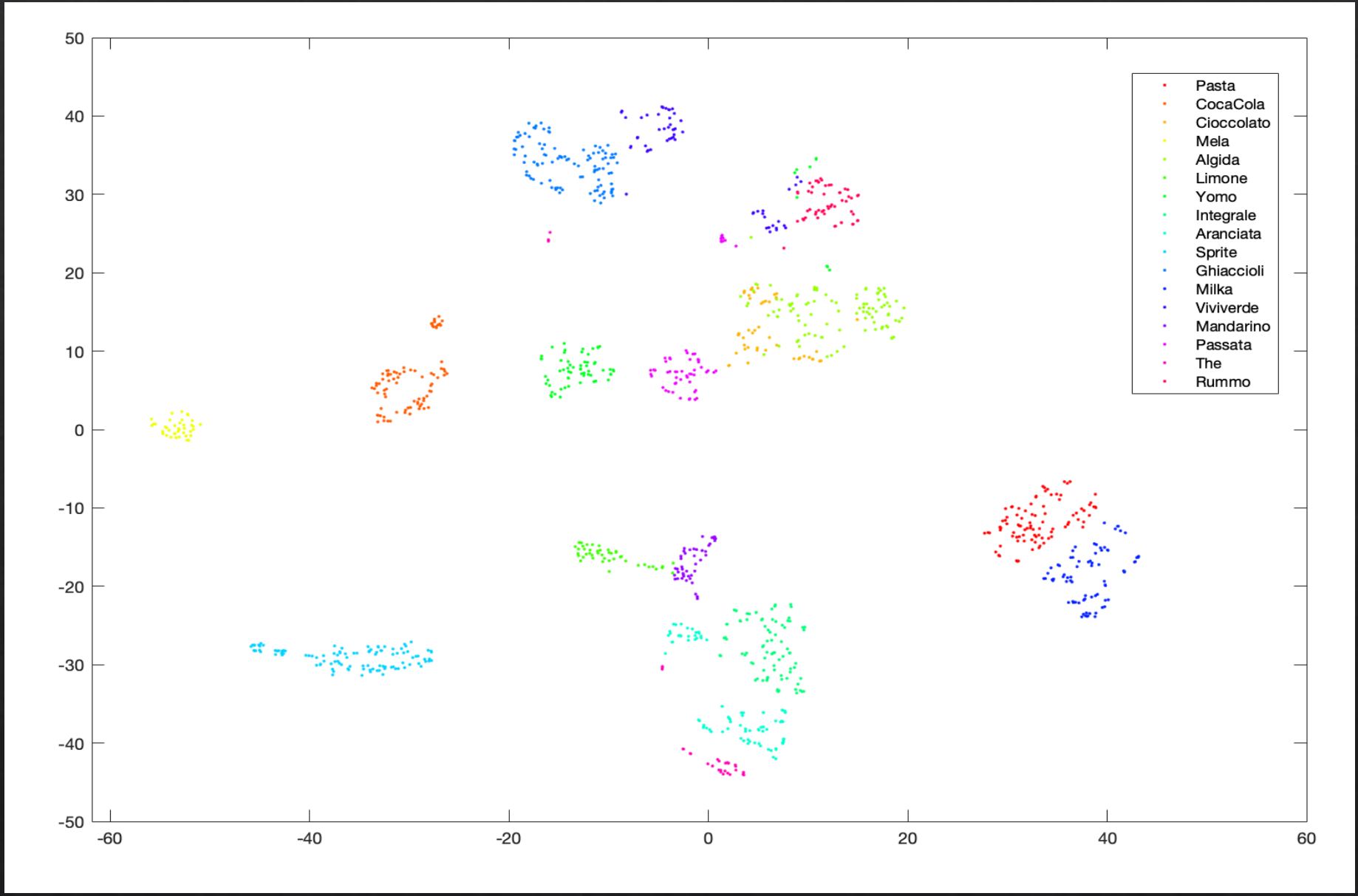
The hu invariant moments are not sufficient to descriminate objects since a lot of them have the same shape.

We didn't implement LBP as invariant for rotation which adds a new problem with the classification

Performance

RELEASE 2:

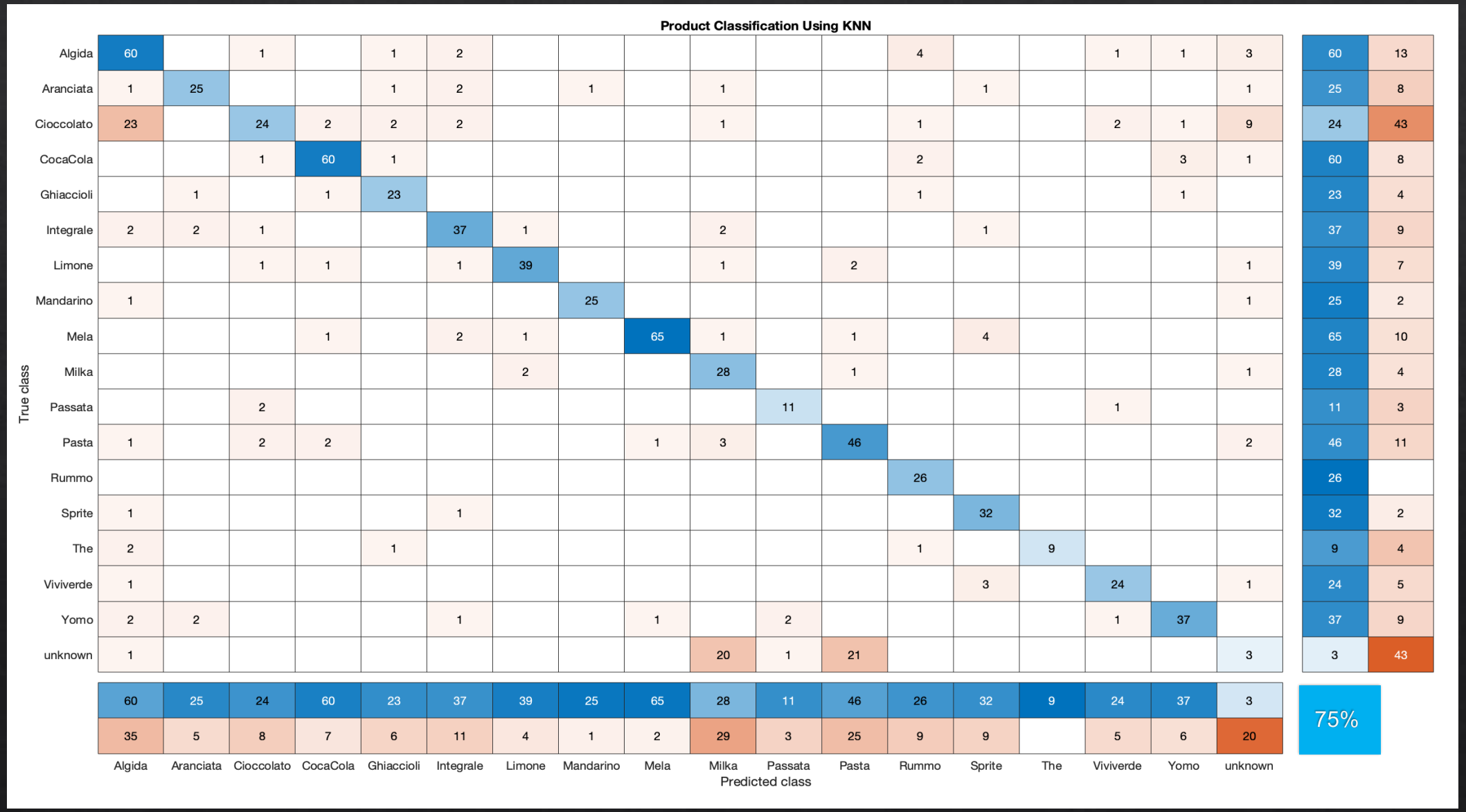
- NO touching objects or slightly overlapping objects
- Standing objects or slightly prospective (added new photos in dataset)
- White balancing
- Feature [hu, CEDD] (LBP and meanColor absorbed in CEDD)



Performance

RELEASE 2:

- NO touching objects or slightly overlapping objects
- Standing objects or slightly prospective (added new photos in dataset)
- White balancing
- Feature [hu, CEDD] (LBP and meanColor absorbed in CEDD)



Performance

RELEASE 2:

- NO touching objects or slightly overlapping objects
- Standing objects or slightly prospective (added new photos in dataset)
- White balancing
- Feature [hu, CEDD] (LBP and meanColor absorbed in CEDD)

We can see a clear improvement in object classification thanks to CEDD descriptors (see tsne shown before).

We're still discriminating objects with color and texture but they're both incorporated inside CEDD descriptors (that also uses edges).

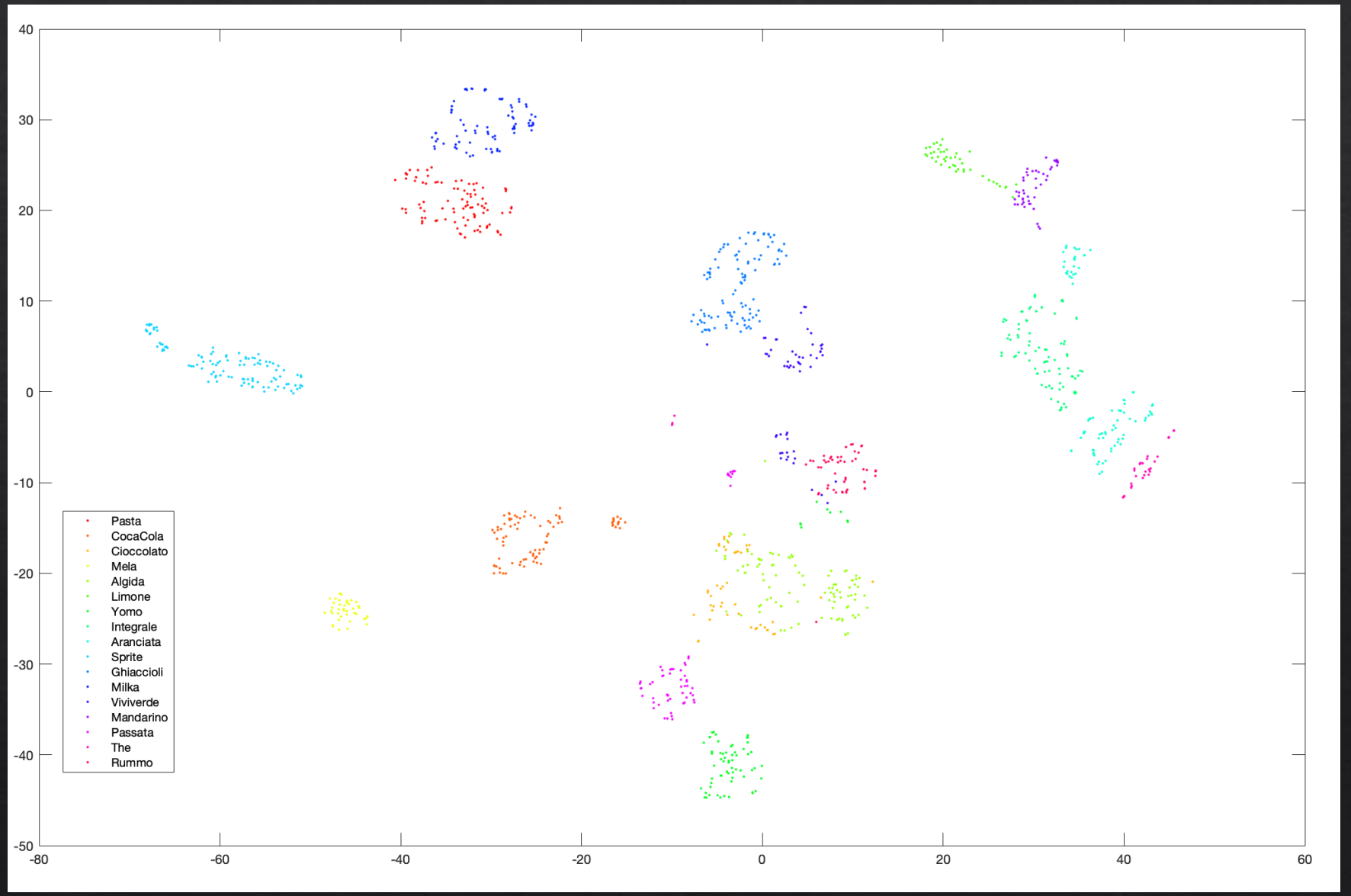
We also took new pictures to handle standing objects.

We still have some classification errors.

Performance

RELEASE 3:

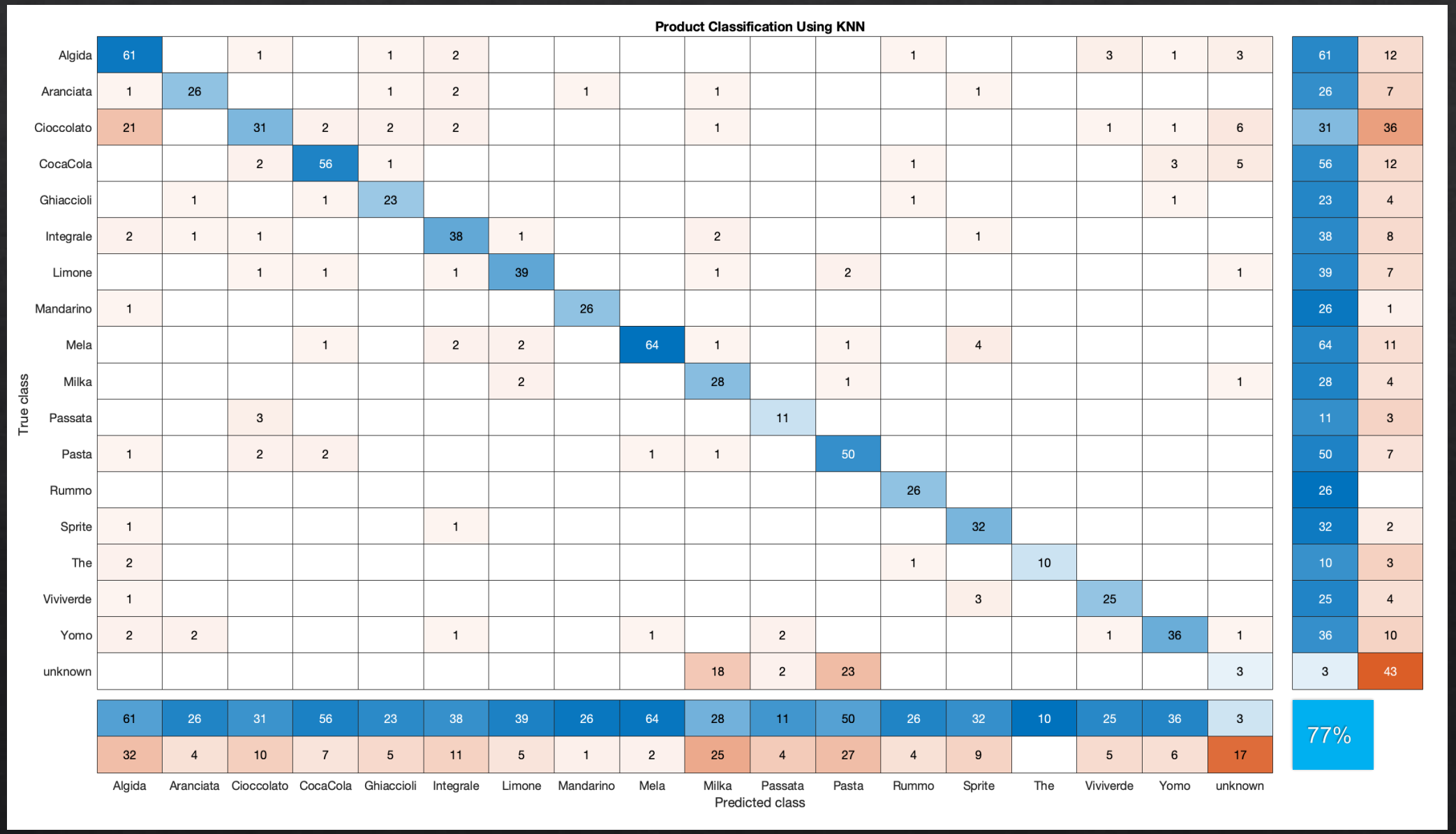
- Touching objects or slightly overlapping objects
- Standing objects or slightly prospective (added new photos in dataset)
- White balancing
- Feature [hu, compactness, meanColor, CEDD] (We tried more invariant features)



Performance

RELEASE 3:

- Touching objects or slightly overlapping objects
- Standing objects or slightly prospective (added new photos in dataset)
- White balancing
- Feature [hu, compactness, meanColor, CEDD] (We tried more invariant features)



Performance

RELEASE 3:

- Touching objects or slightly overlapping objects
- Standing objects or slightly prospective (added new photos in dataset)
- White balancing
- Feature [hu, compactness, meanColor, CEDD] (We tried more invariant features)

We've improved our classifier by introducing two new features:

Mean color (previously discarded) and compactness.

We can see an improvement of performance by 2%.

We also introduced the possibility of slightly touching objects thanks to the watershed transform that doesn't add complication in recognition since it just divides two touching objects

Algorithm considerations

Pros:

- Robust features.
- Classification invariant for rotation, translation or scale.
- Light invariant (if homogeneous).

Cons:

- Confusion in classifying products with similar colors.
- Unknown class often fails (we could have imposed different weight on each feature vector or using a threshold on the euclidean distance during KNN).
- Not homogeneus illumination can causes issues.
- Because of the presence of standing objects, products with similar shapes and colors are often misclassified.

Future improvements

- Delete shadows using color space different from RGB (like channel H/S in HSV color space)
- Improve unknown class
- Automatically find paper borders and crop image by selecting strong magnitude edges.
  If an object is standing on the paper border the edge of the paper is cut.
  That could be easily adjusted by using a Hough transform (edge_linking).

Group involvement

- Sorrentino Alessandro (95%):
  - Dataset creation
  - Main classification algorithm logic & pipeline (all scripts inside project)
  - Segmentation
  - Feature extraction
  - Classificator trainer
  - Other project utility scripts (all utility scripts including white_balancing)
  - Documentation

Group involvement

- Zuccarella Stefano (80%):
  - Dataset creation
  - Watershed algorithm
  - Color balancing
  - Classificator trainer
  - Feature extraction
  - Documentation

Group involvement

- Ratti Burt (70%):
  - Dataset creation
  - Performance analysis
  - Classificator trainer
  - Other project utility scripts
  - Documentation

Bibliography

- Lesson slides
- Digital Image Processing, 4Th Edition, Rafael C. Gonzalez
- CEDD paper: https://pdfs.semanticscholar.org/1af5/293e7e270d35be5eca28cd904d1e8fc9219c.pdf
- Watershed based algorithm: https://blogs.mathworks.com/steve/2013/11/19/watershed-transform-question-from-tech-support/
- Various Matlab documentations