PROJECT EXAMPLE
Coin Detector

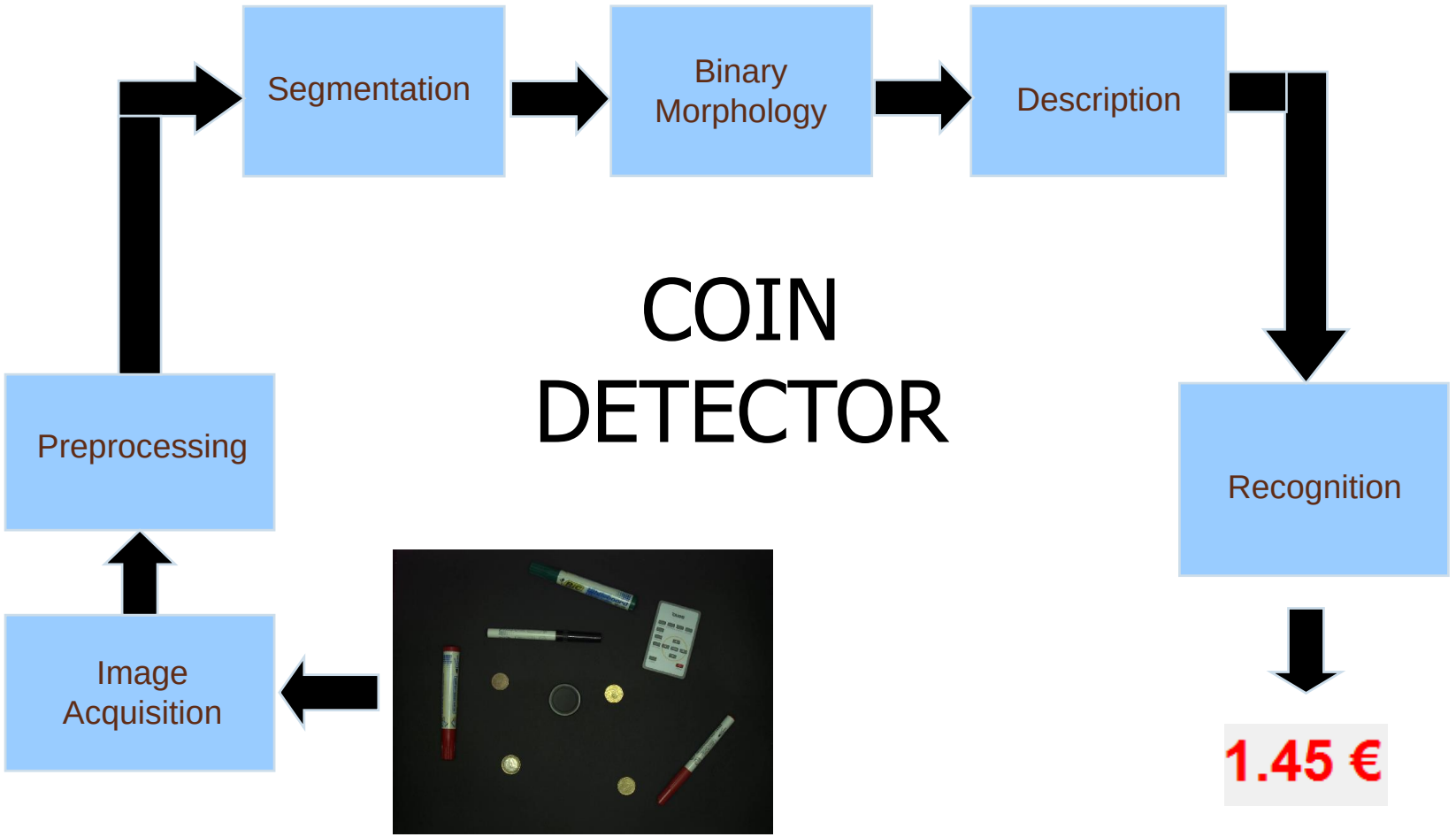
LABORATORY SESSIONS

EXAMPLE – Coin detector

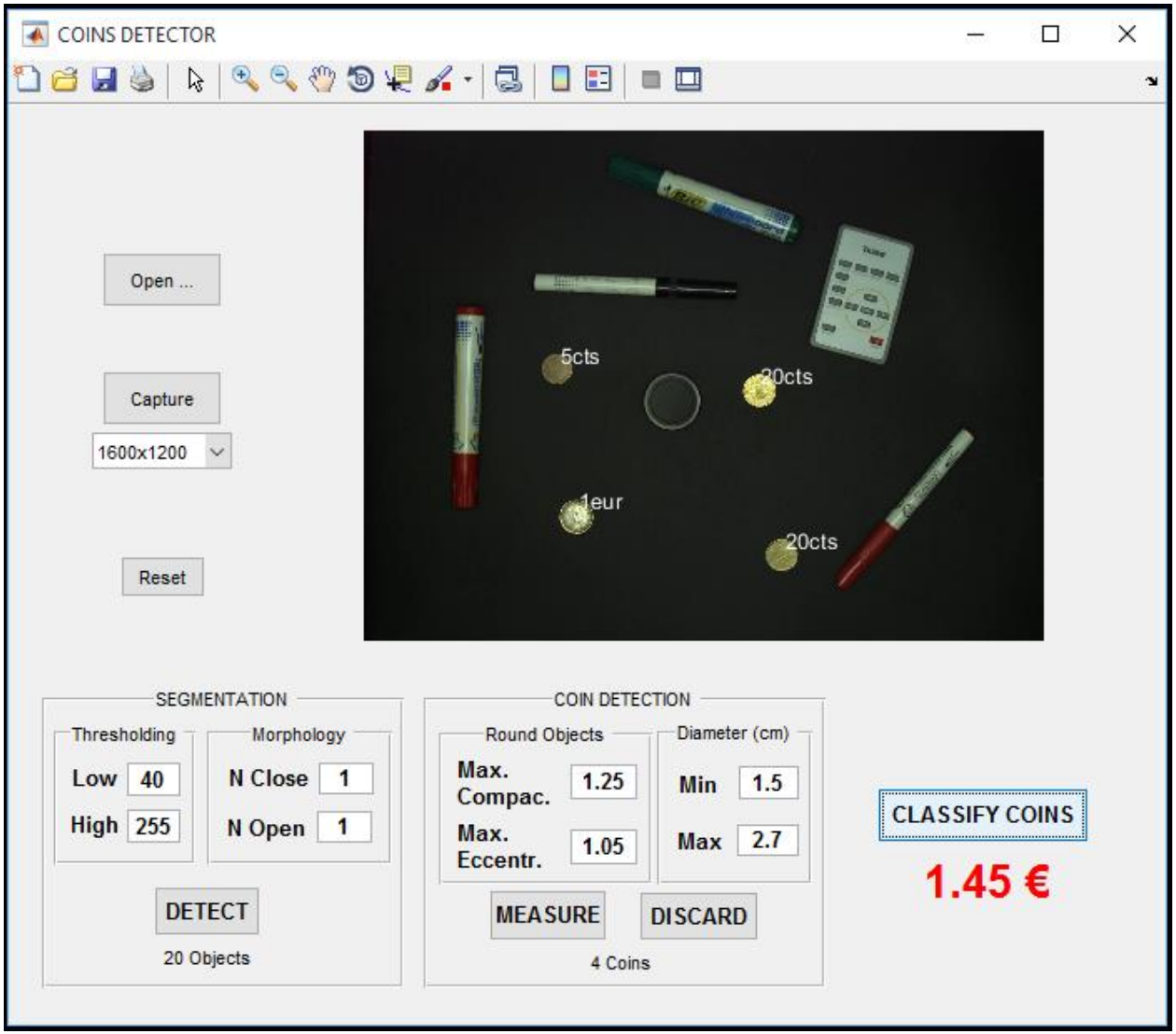
- Example: Coin detector
  - Detection of objects in a scene.
  - Computation of object features and discrimination of round shapes.
  - Classification of coins by size and color.

- Test of algorithm robustness:
  - Performance with low image resolution

EXAMPLE – Coin detector

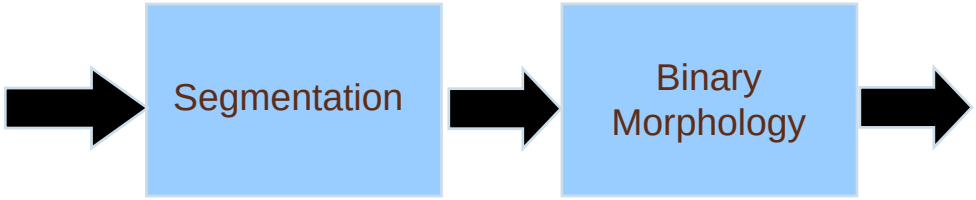


EXAMPLE – Coin detector

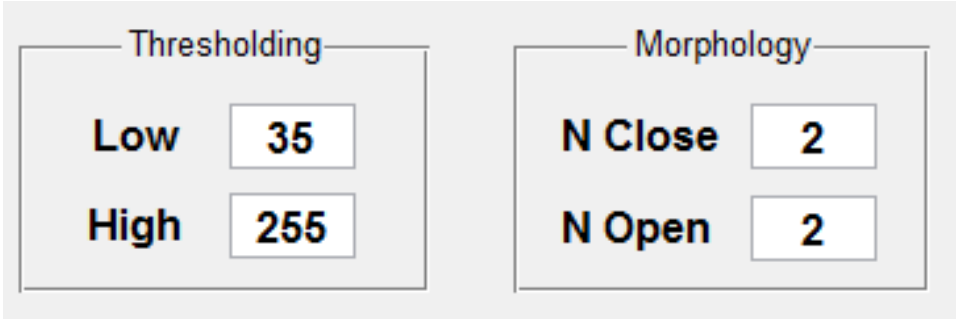


EXAMPLE – Coin detector

1st Step: Object Detection

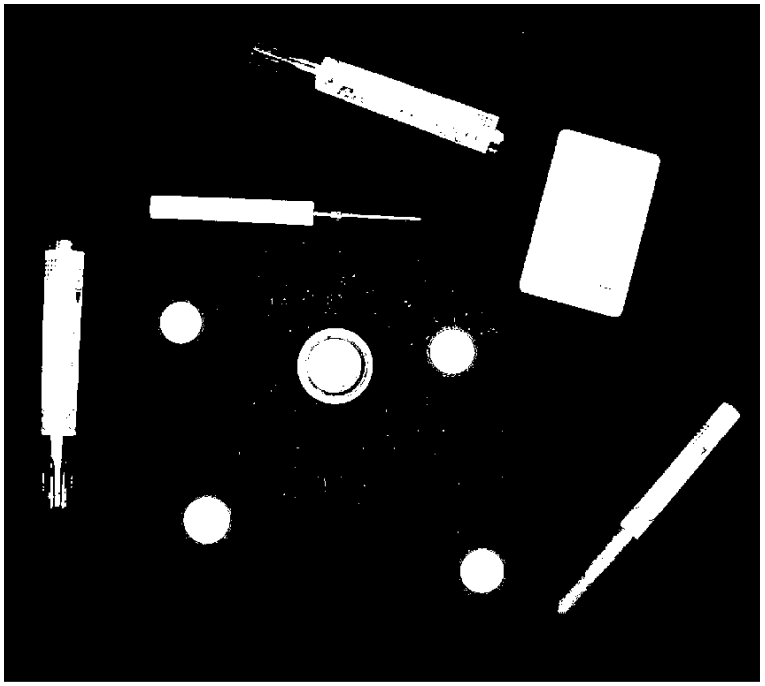


- Thresholding applied to gray level
- Binary morphology: close and open



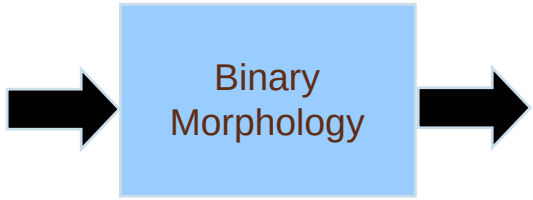
EXAMPLE – Coin detector

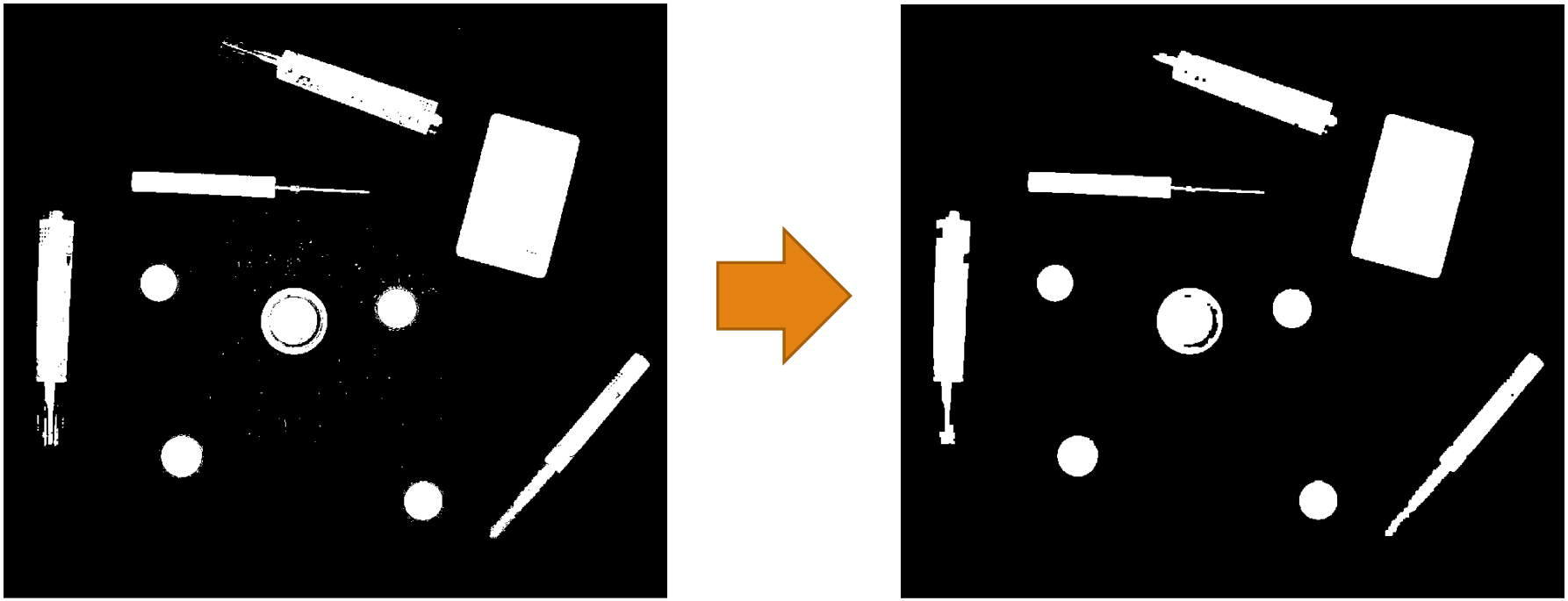
1st Step: Object Detection



EXAMPLE – Coin detector

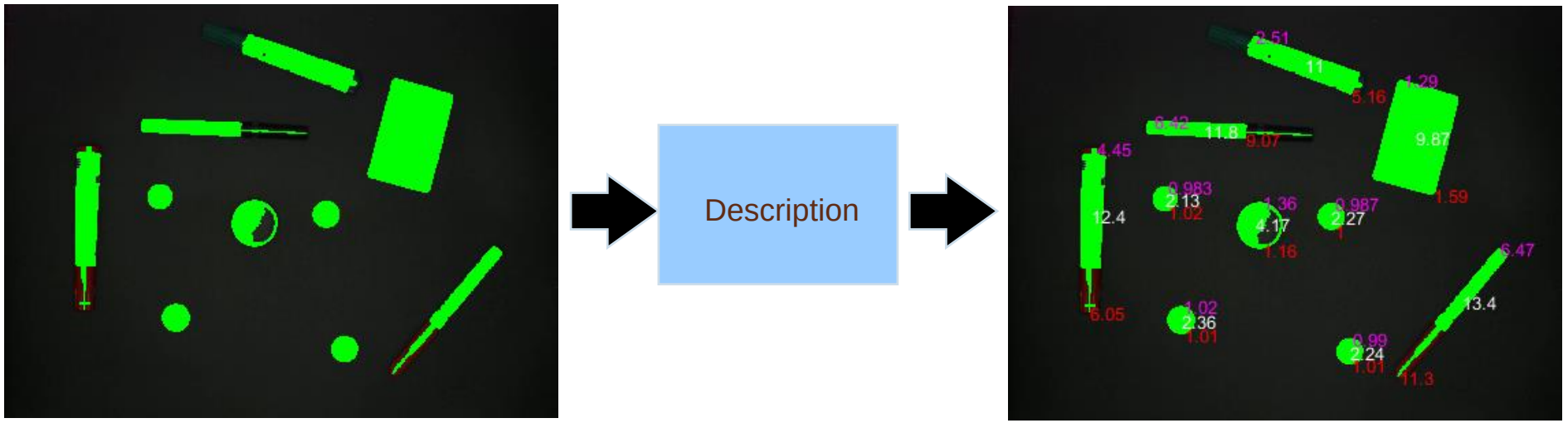
1st Step: Object Detection





EXAMPLE – Coin detector

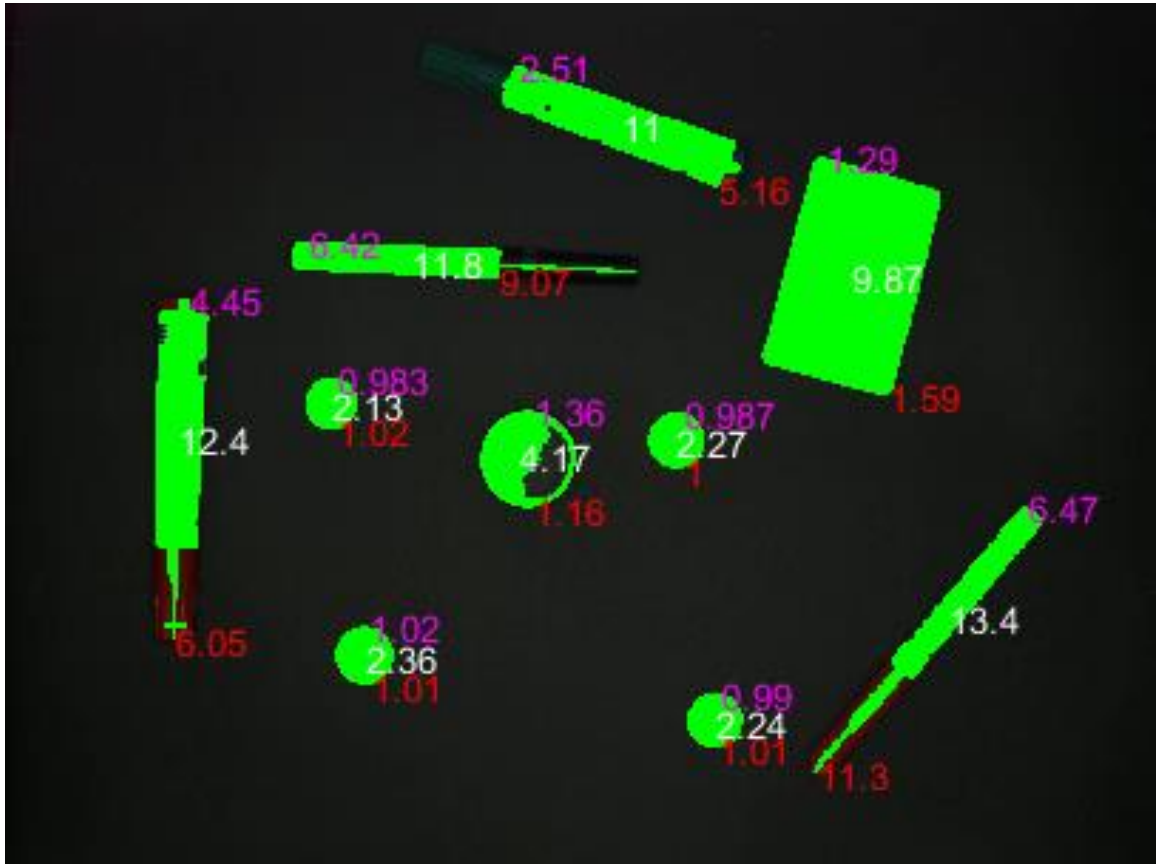
2nd Step: Object Description



- Description of objects to identify coins
- Discrimination of round objects by its compactness (magenta), eccentricity (red) and diameter (white)

EXAMPLE – Coin detector

2nd Step: Object Description



$$\text{Compactness} = \frac{p^2}{4\pi A}$$

$$\text{Eccentricity} = \frac{\text{Major Axis}}{\text{Minor Axis}}$$

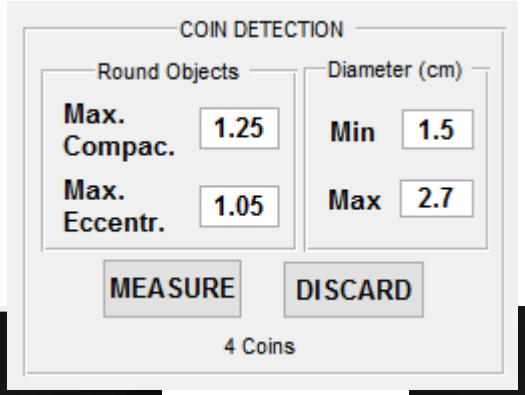
Round object

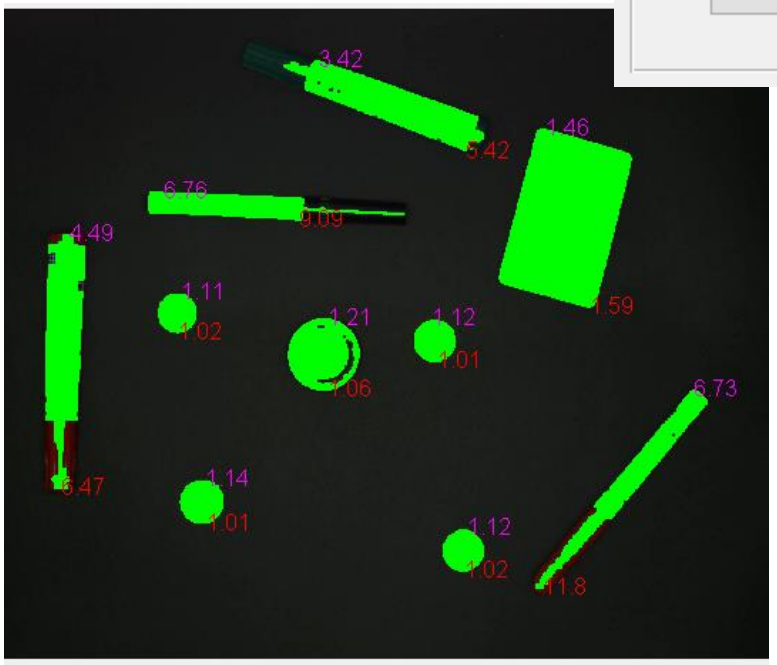
Eccentricity = 1

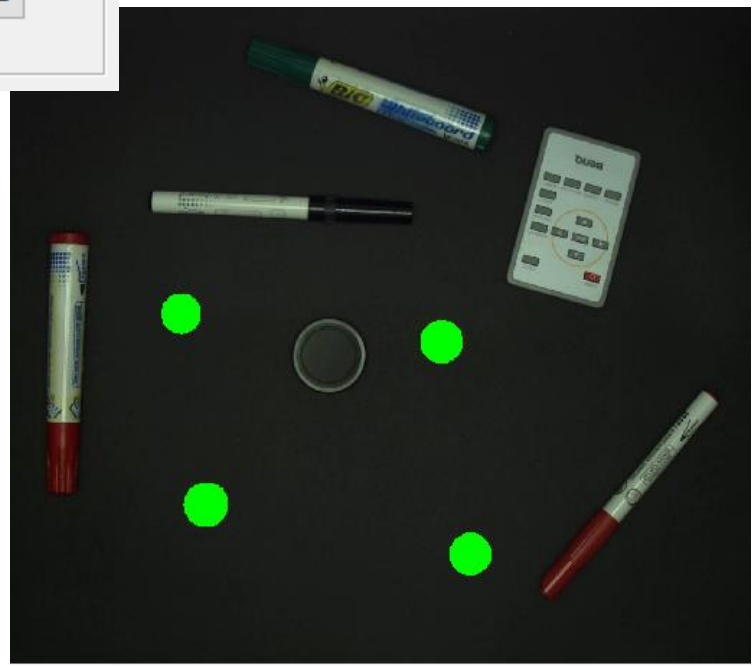
Compactness = 1

EXAMPLE – Coin detector

3rd Step: Discard non-round and smaller/larger objects

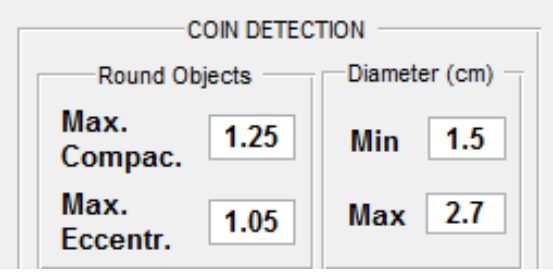






EXAMPLE – Coin detector

3rd Step: Discard non-round and smaller/larger objects



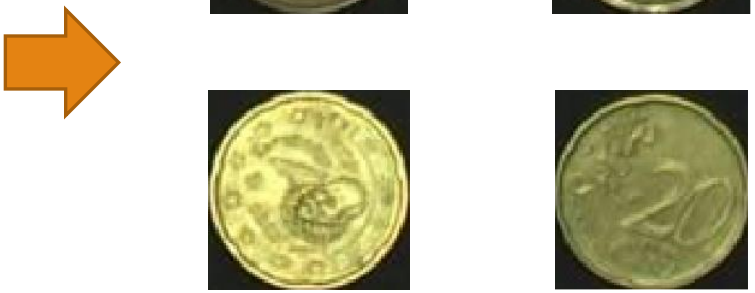
How to compute region features in Matlab:

```
objects=bwconncomp(image_bw);

regionprops(objects,'Area');
                    'Perimeter'
                   'MinorAxisLength'
                   'MajorAxisLength'
```

EXAMPLE – Coin detector

4th Step: Classify coins (by size)



EXAMPLE – Coin detector

4th Step: Classify coins (by size)

2 Euro - 25.75 mms.

1 Euro - 23.25 mms.

50 Cents - 24.25 mms.

20 Cents - 22.25 mms.

10 Cents - 19.75 mms.

5 Cents - 21.25 mms.

2 Cents - 18.75 mms.

1 Cent - 16.25 mms.

EXAMPLE – Coin detector

4th Step: Classify coins (by size)

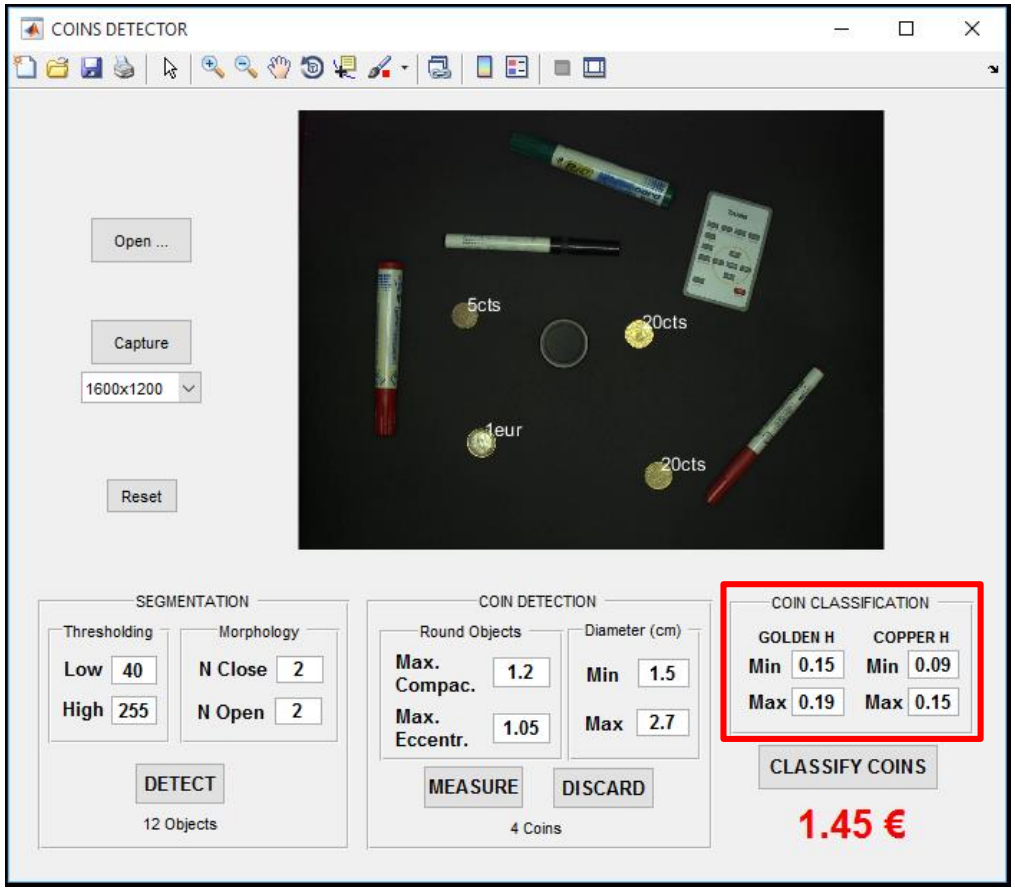
H = 32.2 cm

Conversión (ppp): Pixel ➔ Distance (mm)

EXAMPLE – Coin detector

4th Step: Classify coins (by color)

(*Correction of eventual size errors*)



EXAMPLE – Coin detector

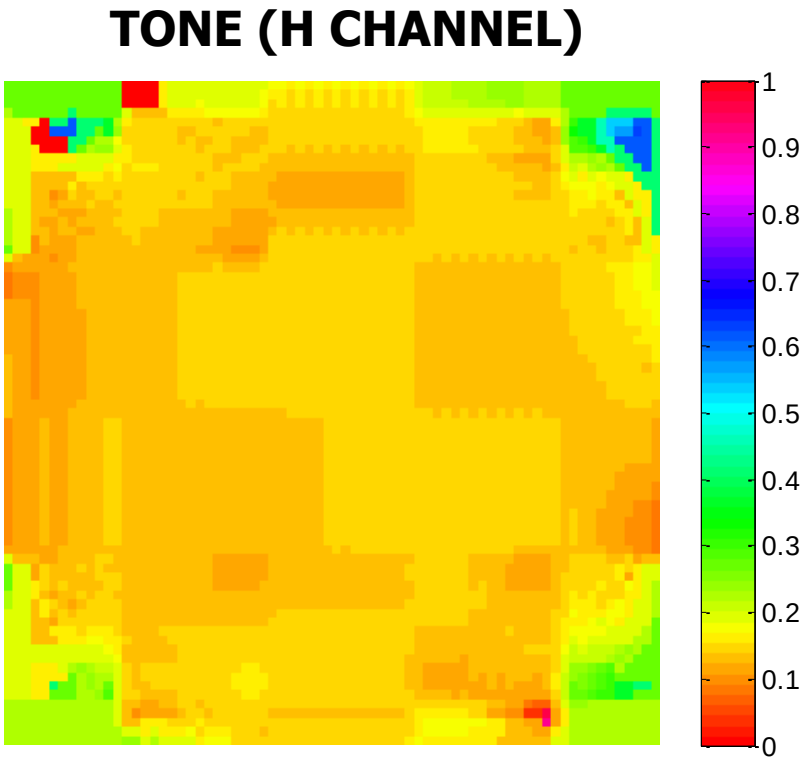
4th Step: Classify coins (by color)

COPPER H

Min 0.09

Max 0.15

(Correction of eventual size errors)



EXAMPLE – Coin detector

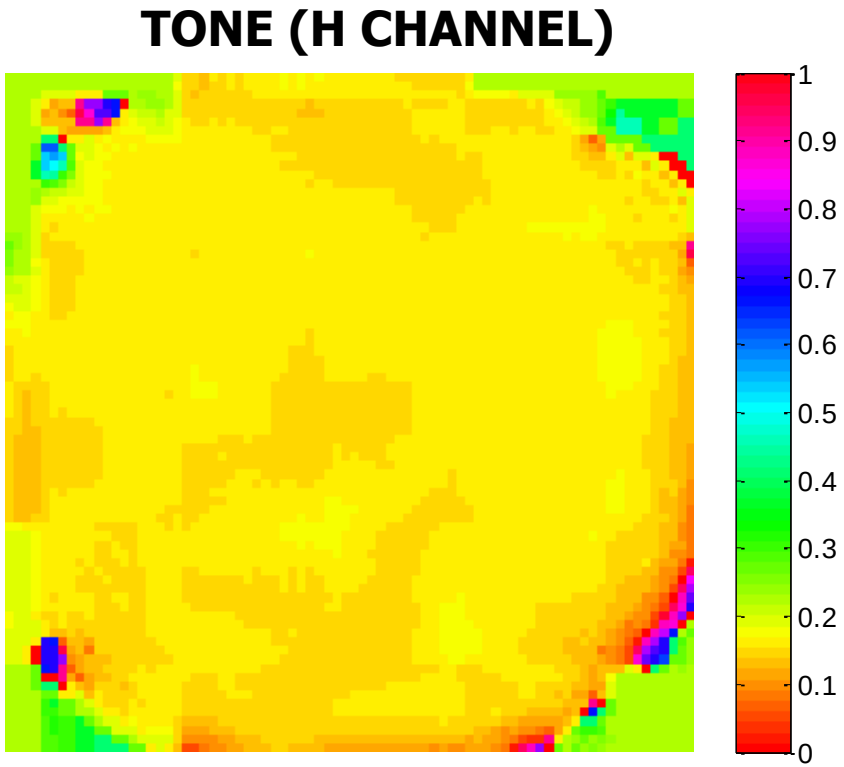
4th Step: Classify coins (by color)

GOLDEN H

Min 0.15

Max 0.19

(Correction of eventual size errors)

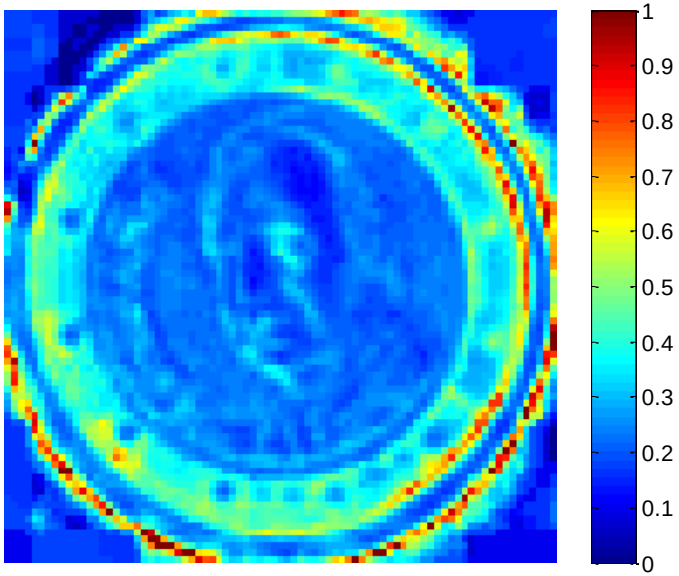


EXAMPLE – Coin detector

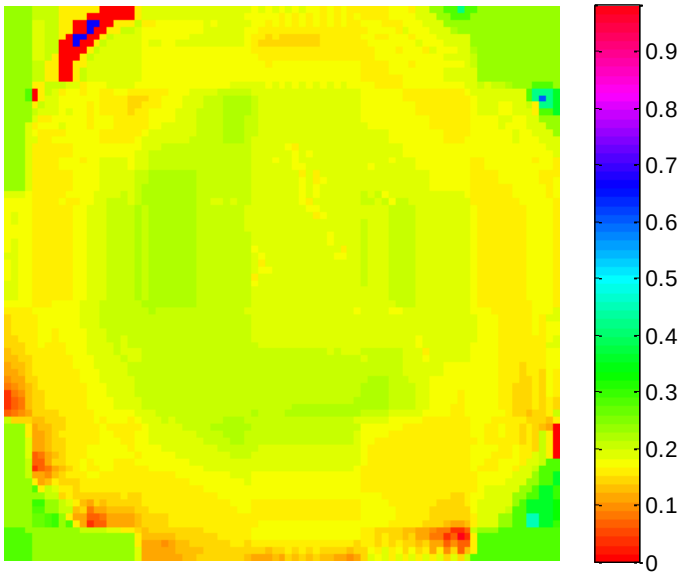
4th Step: Classify coins (by color)

1 & 2 EUROS: Two color zones

SATURATION (S CHANNEL)

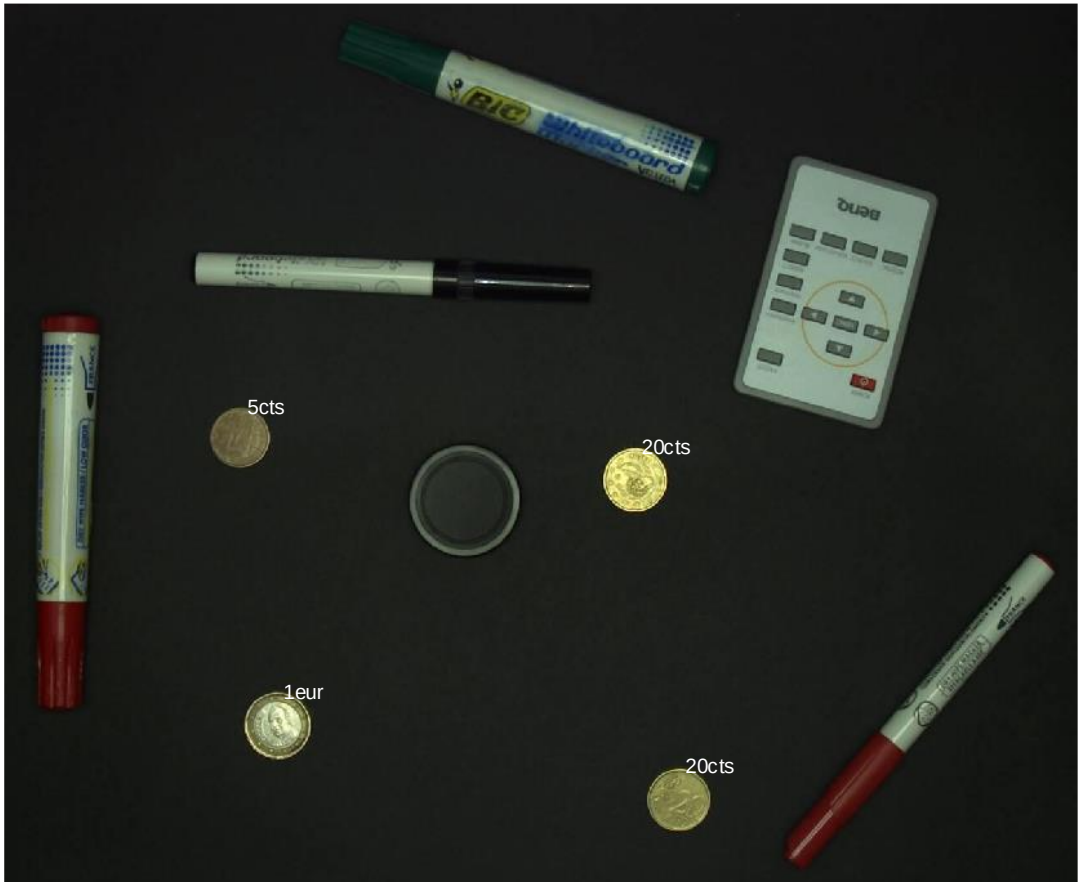


TONE (H CHANNEL)



EXAMPLE – Coin detector

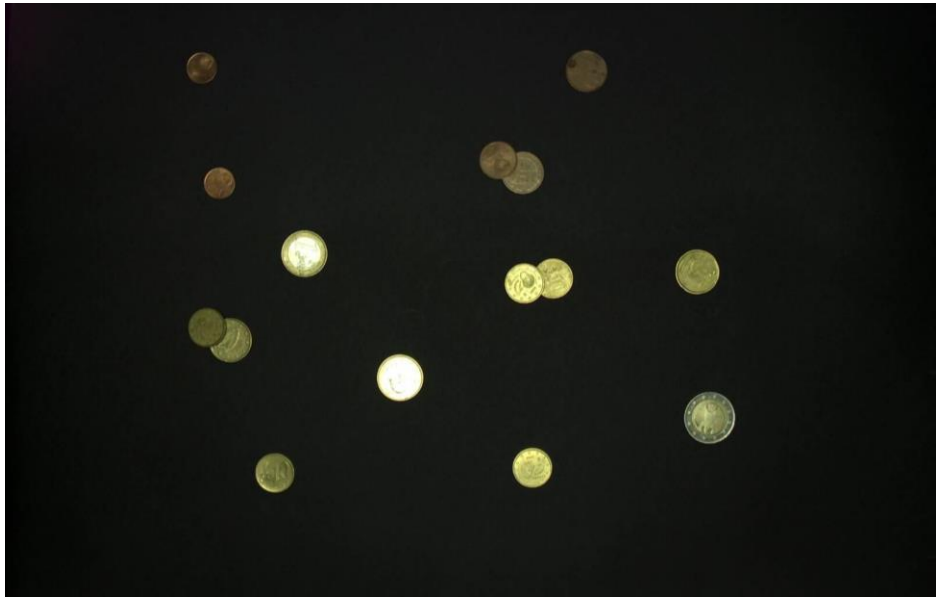
5th Step: Sum of coins' values



1.45 €

EXAMPLE – Coin detector

Solution for overlapped coins????

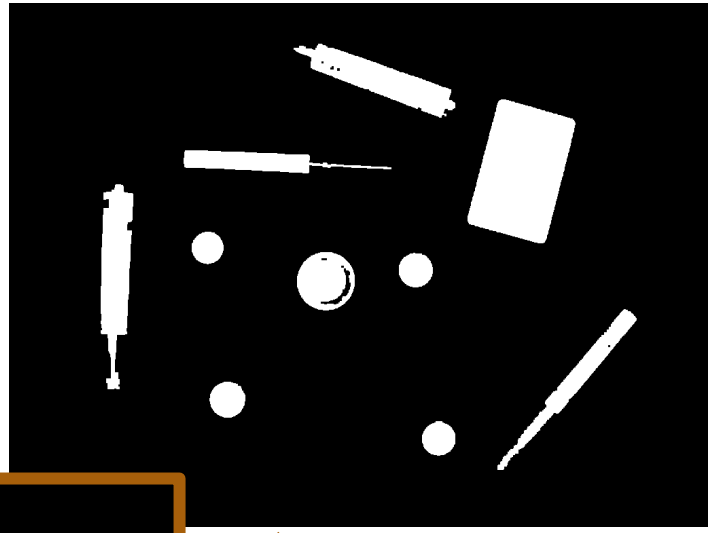


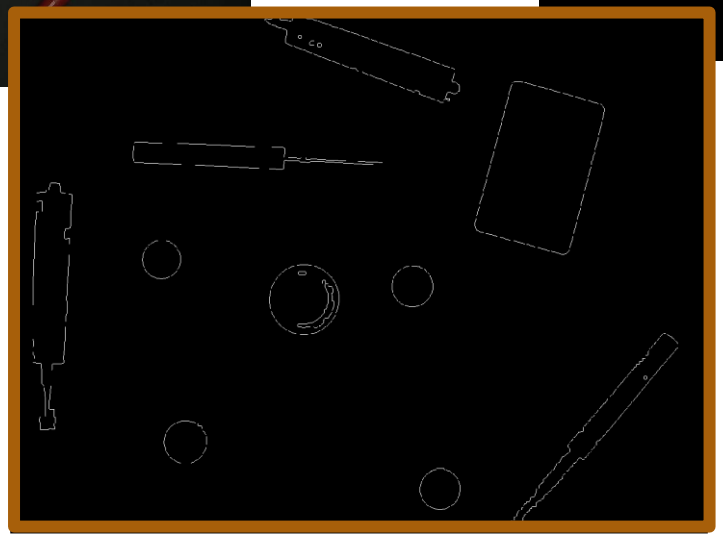
Alternative approach for 2nd and 3rd steps

Hough transform for circles

EXAMPLE – Coin detector

1. Compute the image showing objects' borders





Edge detector

(*edge*)

Binary edge

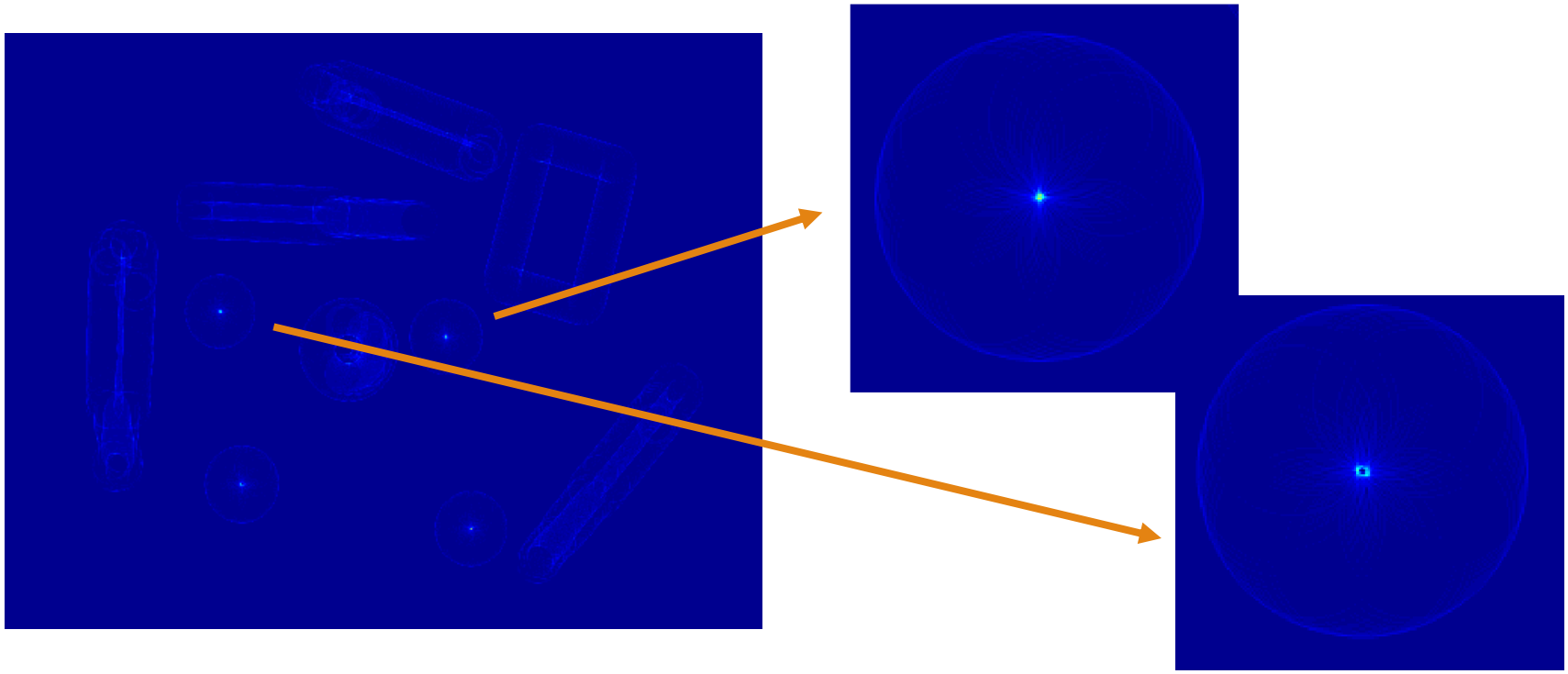
(*bwmorph*)

EXAMPLE – Coin detector

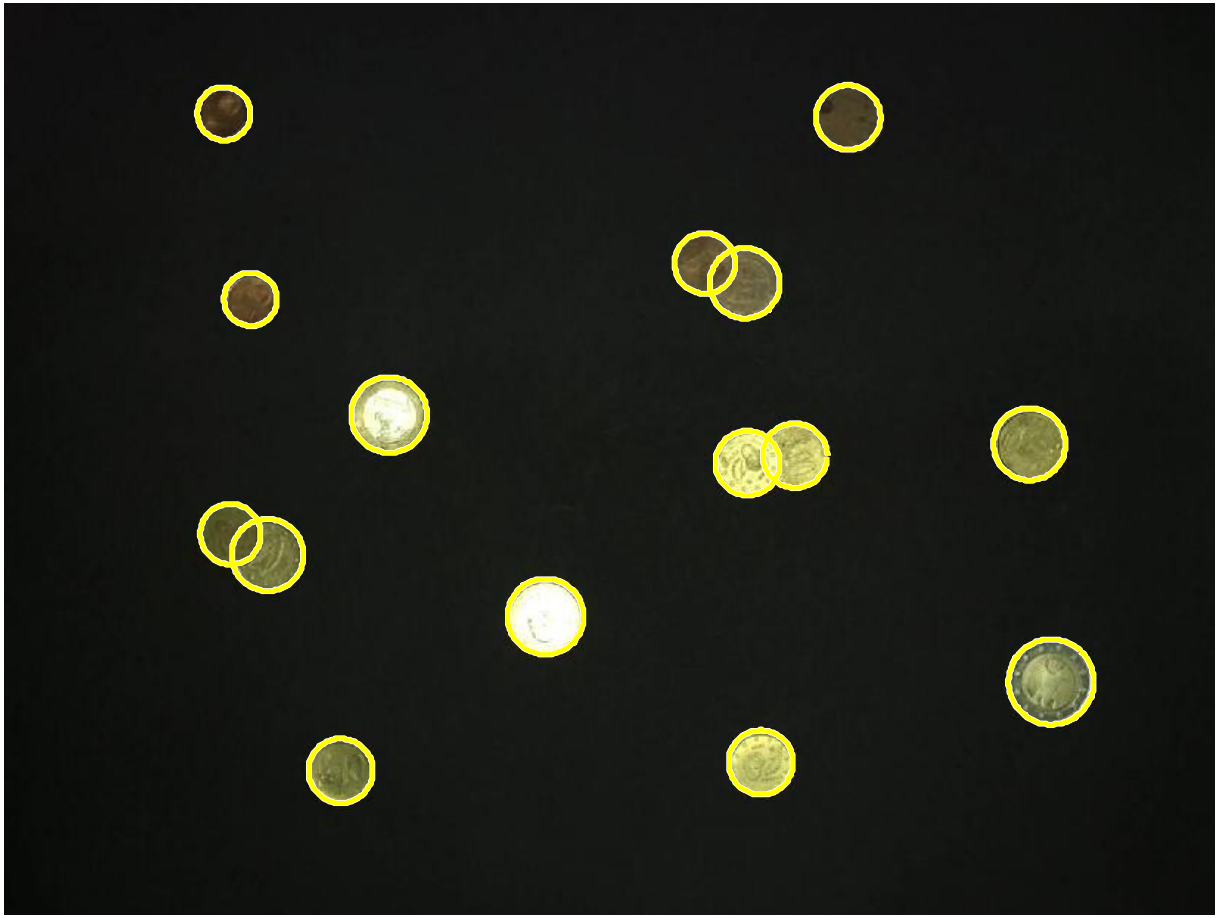
2. Compute the Hough transform

```
[y0detect,x0detect,Accumulator]=houghcircle(image_bord,r0,umb_hough);
```

(Show peaks on possible circumference centers for a given radius)



EXAMPLE – Coin detector



Performance of Hough transform for overlapped coins

EXAMPLE – Coin detector

Improvements:

- Pre-processing:
  - Low-pass filtering to improve segmentation.
  - Camera calibration to avoid measurement inaccuracy near scene edges.
- Segmentation:
  - Local thresholding to correct illumination.
  - Overlap removal by applying binary morphology (alternative to circle Hough transform).