Visual Inspection of Motorcycle Connecting Rods

Computer Vision and Image Processing – Project Work

Francesca Boccardi

Introduction

The aim of the project is to develop a software system aimed at visual inspection of motorcycle connecting rods.

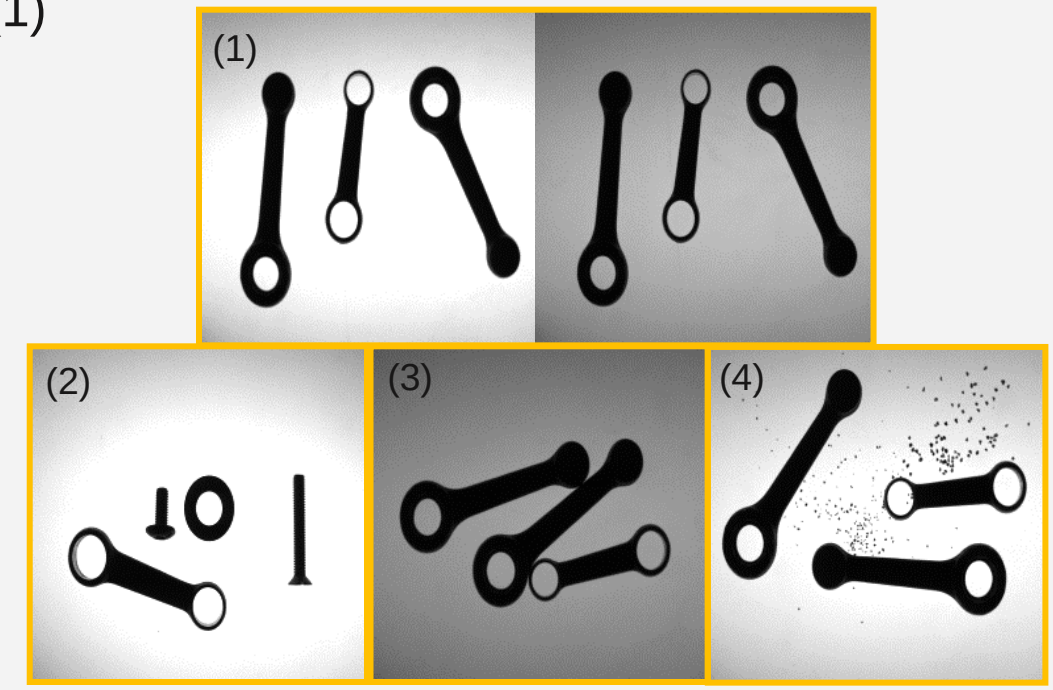
The system must be able to analyze different types of gray-scale images containing connecting rods and provide information about each rod which appears:

- **Type of rod**: one hole (type A), two holes (type B)
- **Position** and **orientation**
- **Length**, **width** and **width at the barycenter**
- Position of the **center** and **diameter of each hole**

Image Features

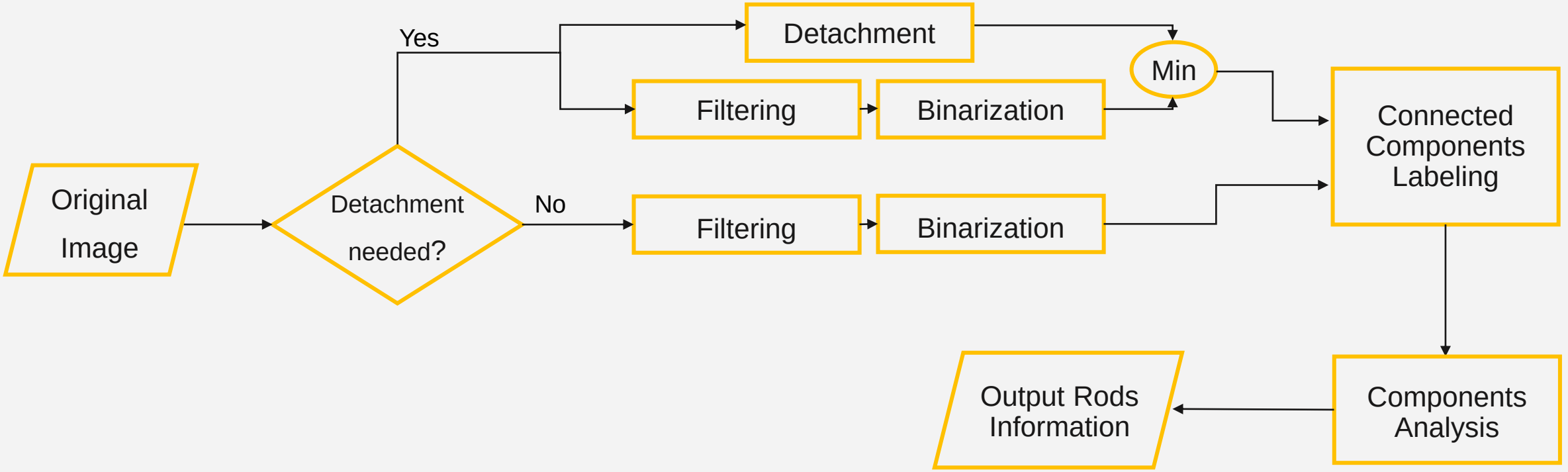
All images have been taken by the backlighting technique, so that rods are easily distinguishable from the background. However, other factors must be considered:

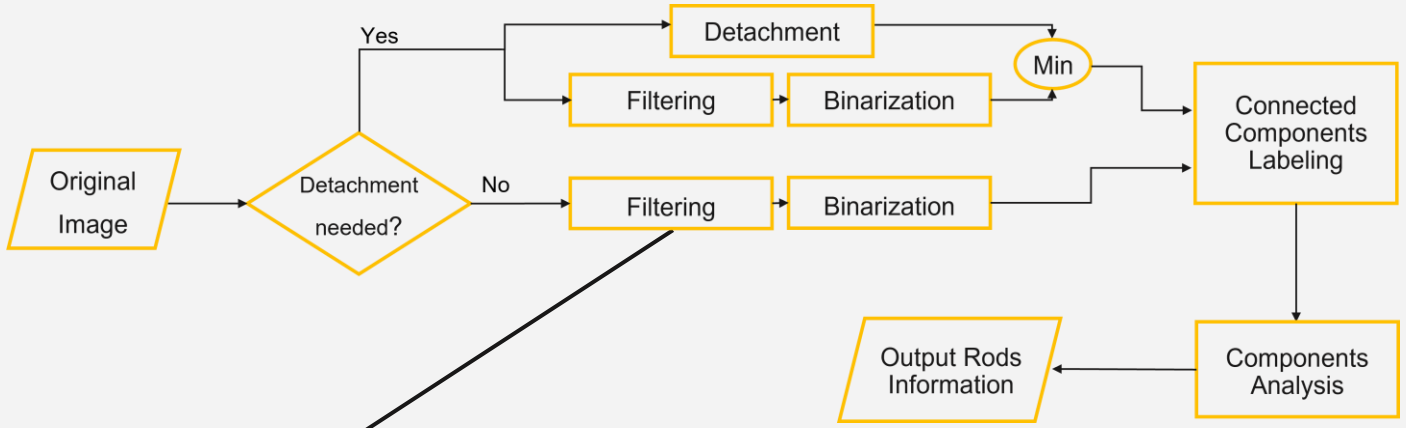
- The **power of the lighting source** may change. (1)
- They may contain **objects different from rods**, which must not be analyzed. (2)
- Rods appearing in the image can have **contact points**. (3)
- The **inspection area may be dirty** due to the presence of scattered iron powder. (4)



System's overview

The entire inspection system can be summarized as follows:



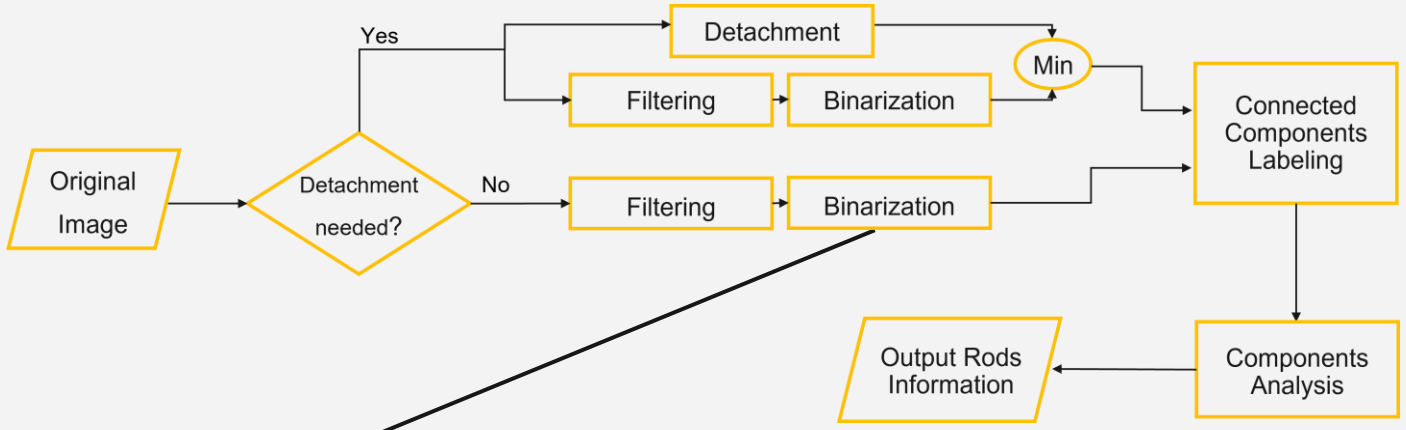


Filtering

Filtering is performed by 5 applications of a median filter with 3x3 kernel.

Original image

Filtered image

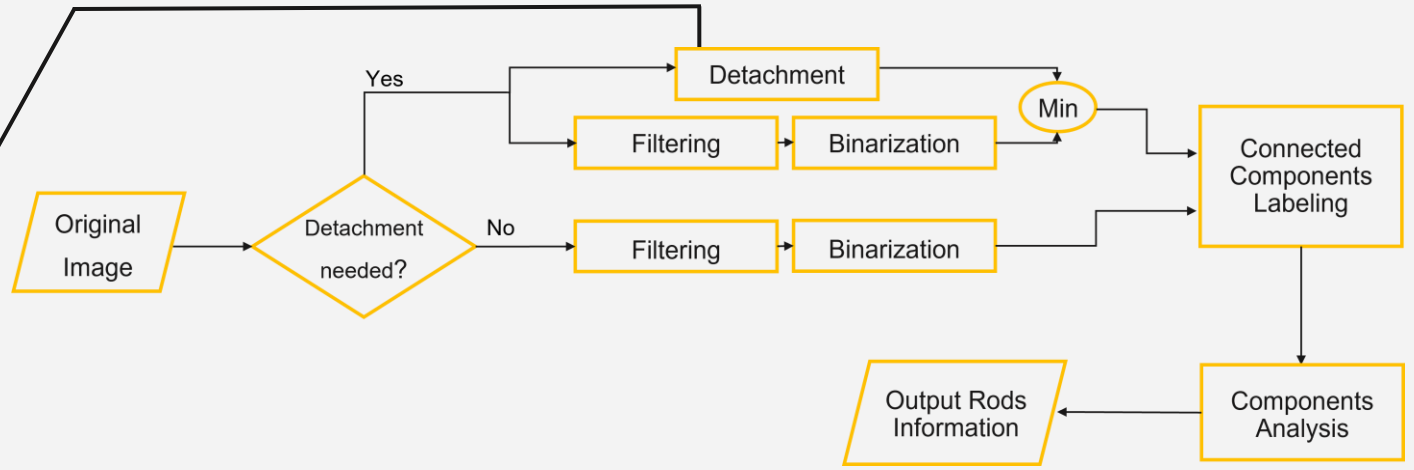


Binarization

The filtered image is then binarized. Due to non-uniform lighting conditions, it was essential to use the Otsu's Algorithm, which performs an automatic selection of the binarization threshold.

Binary image

Detachment



If needed, parallel to image filtering, the original (still unfiltered) image undergoes a «detachment rods» process.

Detachment steps

- **Binarization by Otsu's Algorithm**;
- Connected component labeling;
- For each component having area bigger than 6000 (empiric threshold):
  - Find the external contour;
  - Approximate the contour with a polygonal curve and find the convex hull of it;
  - Find the convexity defect points of the contour with respect to the convex hull;
  - Compute the distance between each pair of defect points and take those closer than 25 (empiric threshold) as contact points;
  - For each pair of contact points, connect them with a black line.

Detachment steps

- Binarization by Otsu's Algorithm;
- **Connected component labeling**;
- For each component having area bigger than 6000 (empiric threshold):
  - Find the external contour;
  - Approximate the contour with a polygonal curve and find the convex hull of it;
  - Find the convexity defect points of the contour with respect to the convex hull;
  - Compute the distance between each pair of defect points and take those closer than 25 (empiric threshold) as contact points;
  - For each pair of contact points, connect them with a black line.

Detachment steps

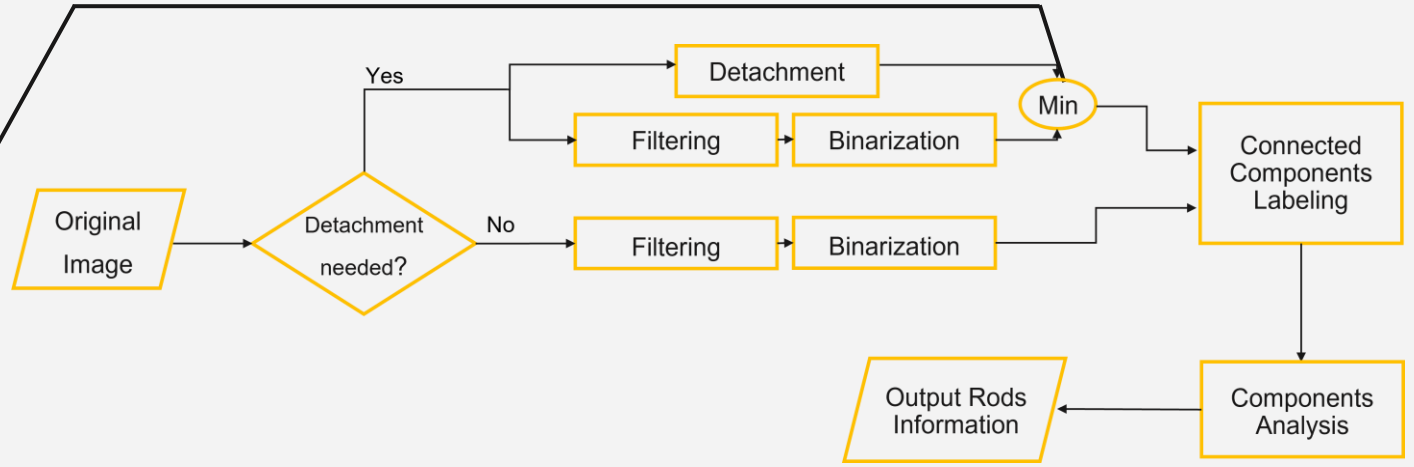
- Binarization by Otsu's Algorithm;
- Connected component labeling;
- **For each component having area bigger than 6000 (empiric threshold):**
  - **Find the external contour;**
  - **Approximate the contour with a polygonal curve and find the convex hull of it**;
  - **Find the convexity defect points of the contour with respect to the convex hull**;
  - Compute the distance between each pair of defect points and take those closer than 25 (empiric threshold) as contact points;
  - For each pair of contact points, connect them with a black line.

Detachment steps

- Binarization by Otsu's Algorithm;
- Connected component labeling;
- For each component having area bigger than 6000 (empiric threshold):
  - Find contour;
  - Approximate the contour with a polygonal curve and find the convex hull of it;
  - Find the convexity defect points of the contour with respect to the convex hull;
  - **Compute the distance between each pair of defect points and take those closer than 25 (empiric threshold) as contact points**;
  - For each pair of contact points, connect them with a black line.

Detachment steps

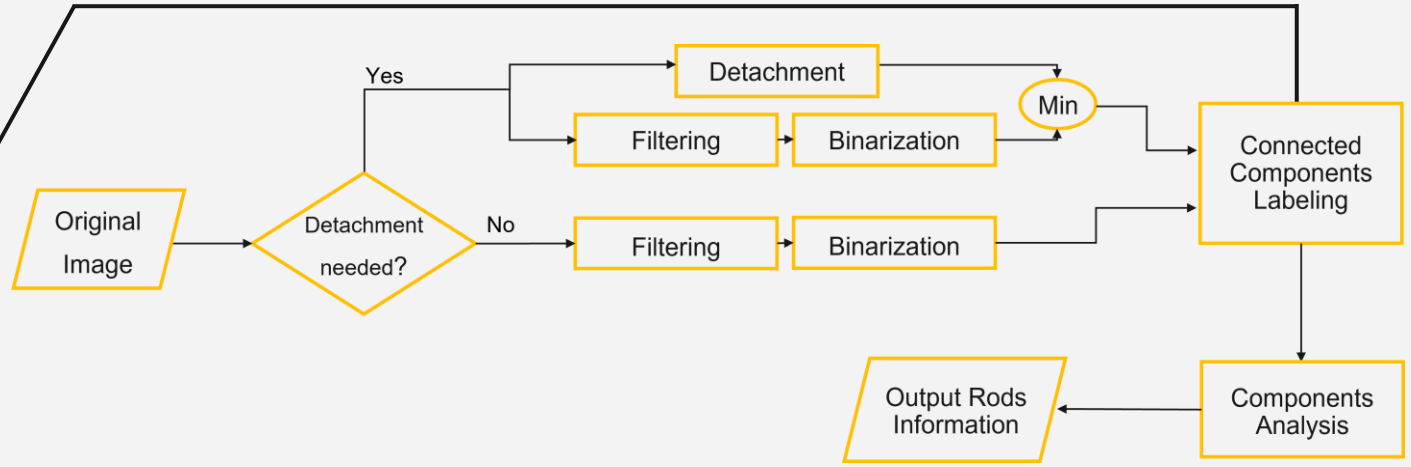
- Binarization by Otsu's Algorithm;
- Connected component labeling;
- For each component having area bigger than 6000 (empiric threshold):
  - Find the external contour;
  - Approximate the contour with a polygonal curve and find the convex hull of it;
  - Find the convexity defect points of the contour with respect to the convex hull;
  - Compute the distance between each pair of defect points and take those closer than 25 (empiric threshold) as contact points;
  - **For each pair of contact points, connect them with a black line.**



Min

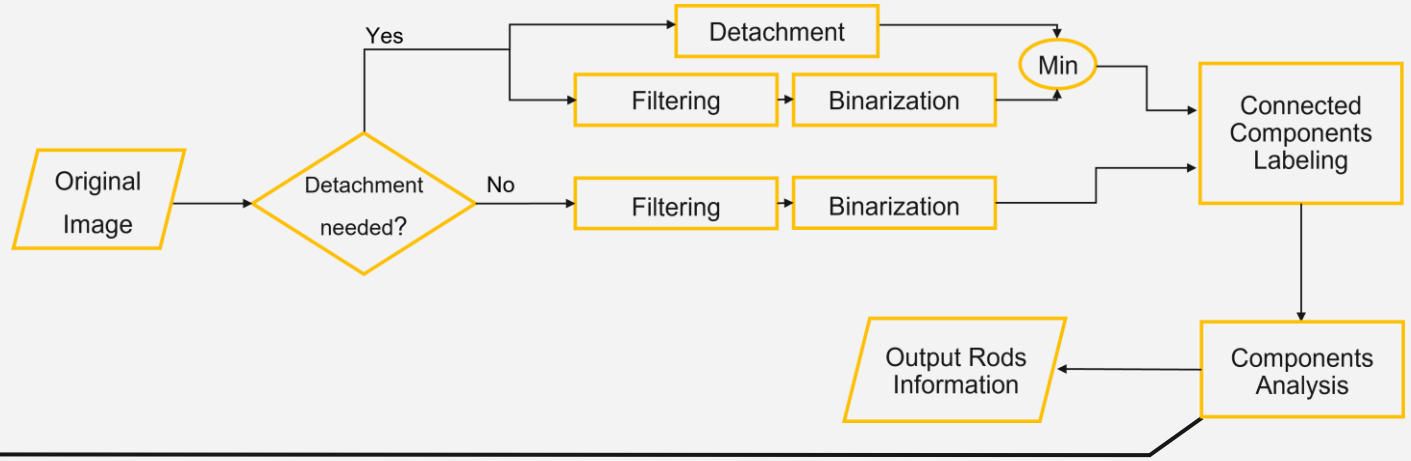
Merging of the binary filtered (attached) image and the binary (unfiltered) detached image through a pixel-wise comparison in order to keep the minimum value between the two.

Connected Components Labeling



On the final binary image, a connected components labeling is performed.

Labeled image



Components Analysis

Each different component is extracted and analyzed. For each of those not been classified as distractors, the system computes all needed features.

Components Analysis
Distractors detection

Two type of distractors may appear:

- **Washers**, detected by looking for components with an haralick circularity value bigger than 9 (empiric threshold)
- **Screws**, detected by looking for components with no holes

Binary image

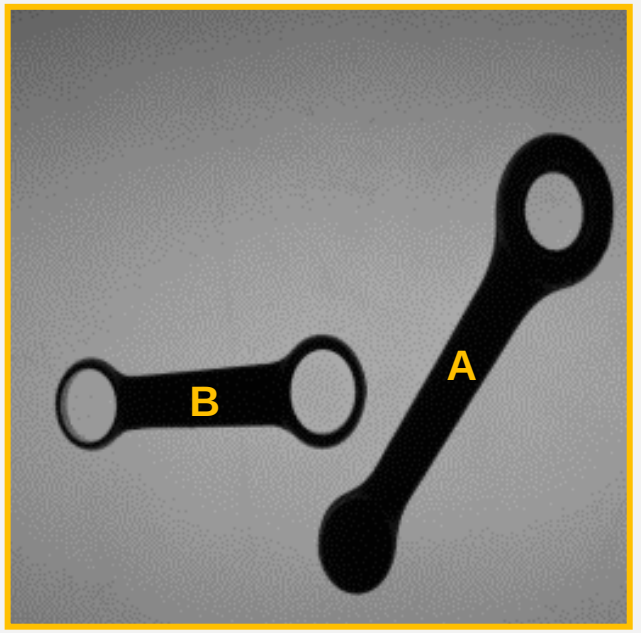
Distractors

Components Analysis
Type of Rod

Classification of the rod as **Type A** or **Type B** relies on the counting of its holes, computed as:

$$\#holes = \#all\_contours - 1$$

where $\#all\_contours$ refers to the total number of rod's contours (external one + holes ones).

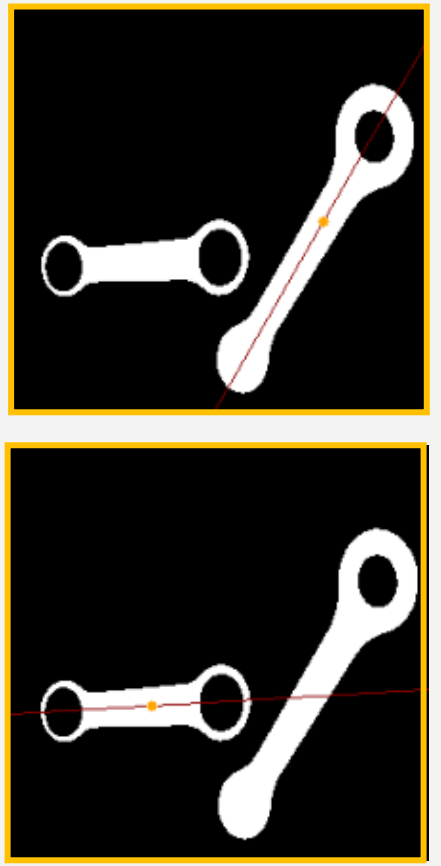


Components Analysis
Position & Orientation

For each rod in the image, the position is found by computing the **centroid** coordinates in terms of the rod's external contour moments:

$$centroid(x, y) = \left\{\frac{M_{10}}{M_{00}}; \frac{M_{01}}{M_{00}}\right\}$$

The **orientation** is instead determined by fitting an ellipse around the rod's external contour and computing its orientation angle.



Centroid and orientation

Components Analysis
Width & Length

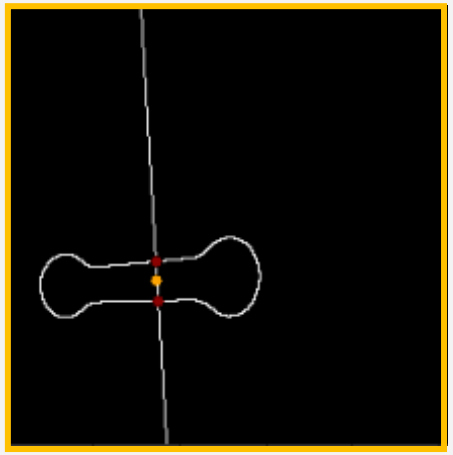
Width and **length** are determined by simply enclosing the rod into a rod-oriented **bounding box** and then computing its dimensions.

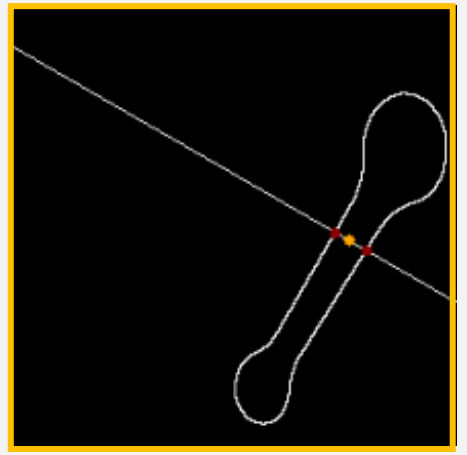
Oriented bounding boxes

Components Analysis
Barycenter Width

Barycenter width is determined by:

- detecting the **intersection points** between the rod's external contour and the line passing through the **barycenter**, perpendicular to the rod's orientation
- computing the distance between the two points so found





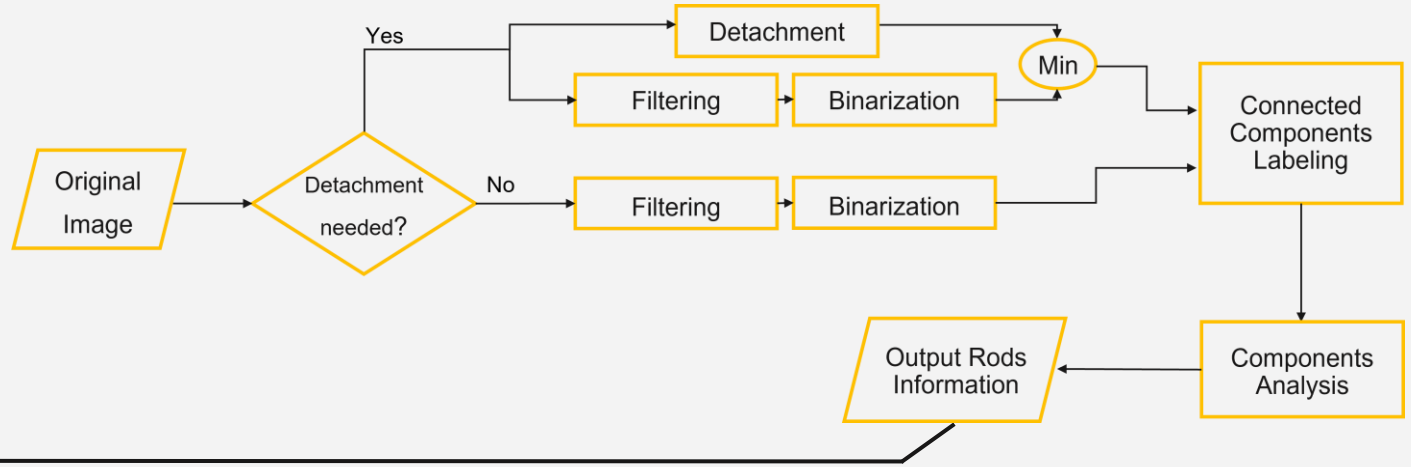
Components Analysis
Holes centers & Diameter

For each hole, its features are computed by:

- Extracting the hole's contour
- Determining the hole's center position by computing the **centroid** coordinates, in terms of the contour's moments

- Computing the **diameter** size as:

$$d = \frac{contour\_length}{\pi}$$

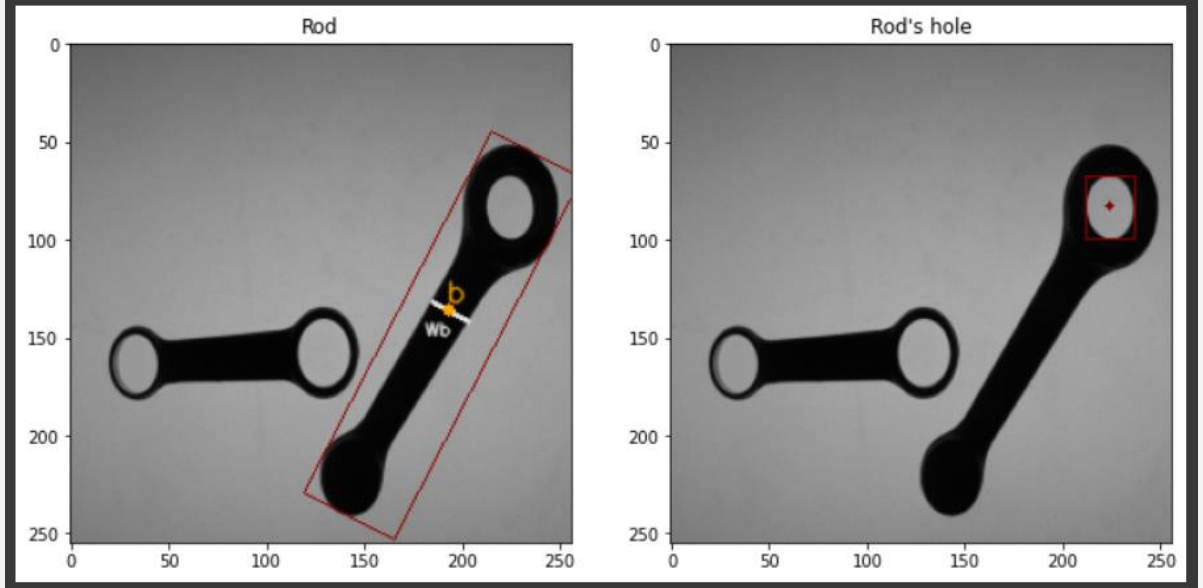


Output Rods Information

All rods features previously computed are then printed and visualized.

Output Rods Information

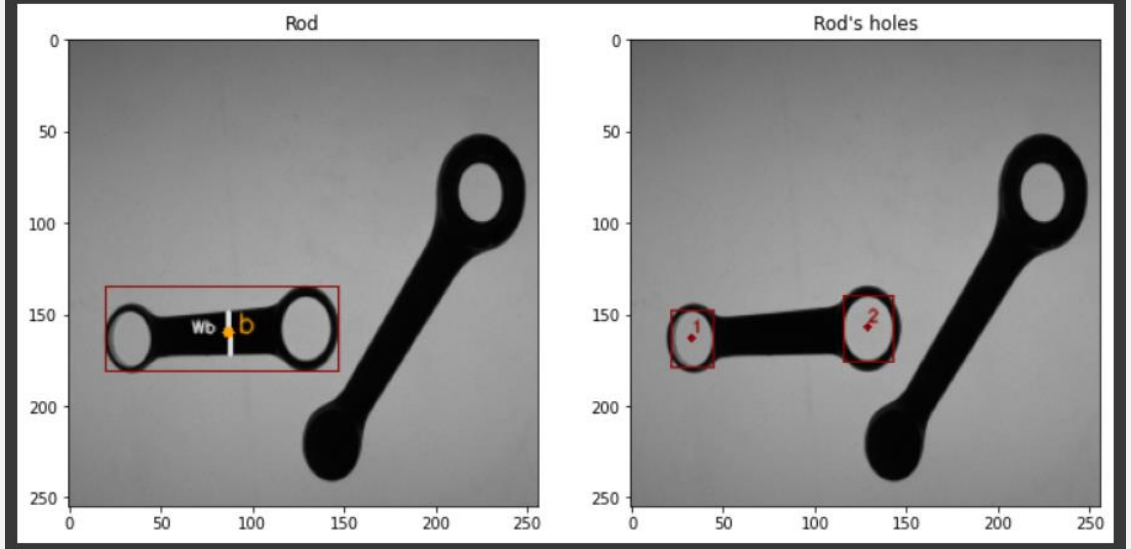
ROD 1




```
Type: A
Position: (193, 136)
Orientation: 60.14°
Lenght: 208.07
Width: 52.11
Width at Barycenter: 20.59

Hole position: (224, 83)
Hole diameter: 30.51
```

ROD 2




```
Type: B
Position: (87, 160)
Orientation: 3.36°
Lenght: 127.0
Width: 46.0
Width at Barycenter: 23.02

Hole_1 position: (33, 163)
Hole_1 diameter: 28.6
Hole_2 position: (129, 157)
Hole_2 diameter: 33.39
```