MOTORVEHICLE
UNIVERSITY OF
EMILIA-ROMAGNA

Alma Mater Studiorum
Università di Bologna

Computer Vision and Image Processing Exam

Motorcycle Connecting Rods Inspection

Francesco Marotta
Simone Maravigna

September 16, 2023

Code Report

The Motorcycle Connecting Rods Inspection Project aims to develop an application capable of conducting an analysis of these rods, providing detailed information such as dimensions, orientation, and rod type.

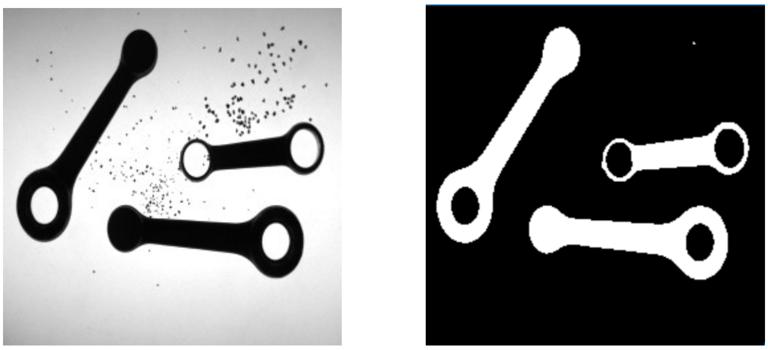
Image Acquisition

The first step in the project was image acquisition from a predefined directory using `glob` and storing them in an array.

Figure 1: Example of an input raw image

Image Preprocessing

The acquired images go through a median filter to remove impulse noise, due to iron powder in some images, `cv2.medianBlur` is applied with a 3x3 kernel and 3 iterations. Images are then binarized using Otsu's Algorithm with `cv2.threshold`. This converts grayscale image to binary representations and inverts the color by making the foreground object white in a black background. We have chosen the Otsu's Algorithm since it is able to automatically threshold the image.

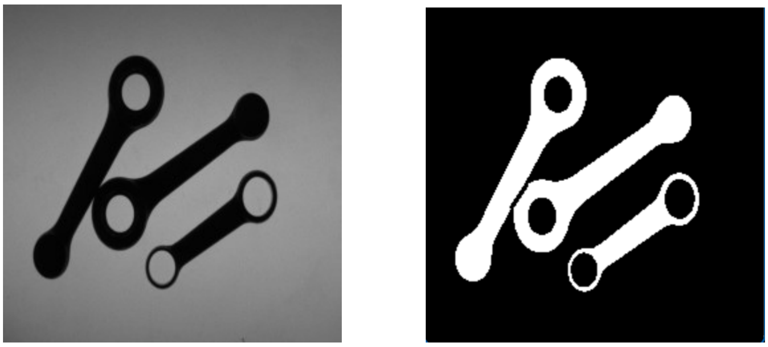


(a) Noisy Input Image (b) Binary denoised Image

Figure 2: Denoising process example

Blob Detection

It is important to detect all the components and analyse them one at a time. In order to do so, first of all, in case of touching components it is mandatory to separate them, and that was implemented by computing the convexity defects using `cv2.convexHull` and `cv2.convexityDefects`. Then two touching elements are divided by a 2px white line between the two nearest convexity defects points. After that, `cv2.connectedComponentsWithStats` is employed to label connected components in the binary images. Each labeled region represents a potential connecting rod.



(a) With touching components (b) With divided touching components

Figure 3: Divide Touching Components

Hole Detection

Once components are detected, the focus shifts to holes detection. To locate holes `cv2.connectedComponentsWithStats` is used again, but this time to the inverted object mask. This provides the holes center and, by using simple geometry, also diameter. If the object has no holes or more than two, it is discarded because it's not a rod.

Dimension Measurement

Measurements of length, width, orientation angle and width at the barycenter are calculated for each connecting rod. For the first two `cv2.minAreaRect` is used to find the Minimum Enclosing Rectangle (MER) for each connecting rod, while the orientation angle is computed using `cv2.fitLine`, which fits a line to the contour of the rod. The angle between the line and the x-axis is calculated to determine the orientation. The width at the barycenter is found as the distance of the two points of intersection between the minor axis and the contour.

Data Visualization

Finally all the information are showed through prompt and some visual representation by using `cv2.imshow`

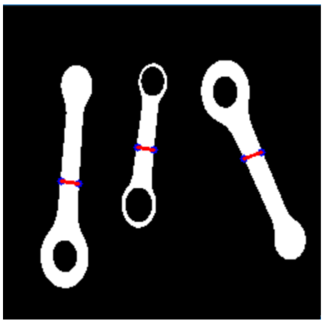
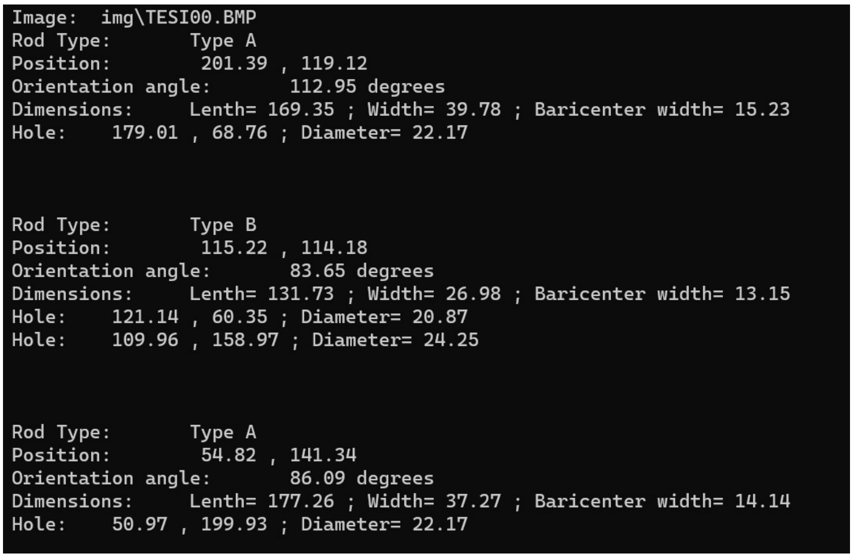


Figure 4: Example of an output image



```
Image:  img\TESI00.BMP
Rod Type:       Type A
Position:        201.39 , 119.12
Orientation angle:       112.95 degrees
Dimensions:     Lenth= 169.35 ; Width= 39.78 ; Baricenter width= 15.23
Hole:    179.01 , 68.76 ; Diameter= 22.17



Rod Type:       Type B
Position:        115.22 , 114.18
Orientation angle:       83.65 degrees
Dimensions:     Lenth= 131.73 ; Width= 26.98 ; Baricenter width= 13.15
Hole:    121.14 , 60.35 ; Diameter= 20.87
Hole:    109.96 , 158.97 ; Diameter= 24.25



Rod Type:       Type A
Position:        54.82 , 141.34
Orientation angle:       86.09 degrees
Dimensions:     Lenth= 177.26 ; Width= 37.27 ; Baricenter width= 14.14
Hole:    50.97 , 199.93 ; Diameter= 22.17
```

Figure 5: Example of output data visualization