Washer Analysis using Image Processing

Tanmay Redkar (2022008)

Student of Department of Electronics Engineering

K. J. Somaiya College of Engineering, Vidyanagar, Vidya Vihar East, Vidyavihar, Mumbai, Maharashtra 400077, India

INTRODUCTION

Nuts, bolts, Washers are the bread and butter of every machine. Our whole life revolves around these simple tools we call fasteners. If any defective washer is used it can cause serious quality control issues. In this paper, I have prepared a python program to automate the process to identify and sort out defective washers. This has been achieved using Image processing tools such as Thresholding, Morphological operations and mathematical operations such as Circularity.

METHODOLOGY

Several pictures of Washers were captured to be used for analysis. All the pictures were cropped to same sizes in order to maintain uniformity in detection. Thresholding was applied on these pictures to filter out redundant information and obtain edges. However, this wasn't enough as different lighting conditions made it difficult to get the edge-to-edge extraction of the washer. Thus, Morphological operations were performed and circularity of the washer was calculated. All the operations were performed using OpenCV and Numpy libraries in Python.

LITERATURE REVIEW

A. Edge Detection

The aim of performing edge detection in general is to significantly reduce the amount of data in an image, which will be used in the further processing. This step increases the computation speed of the approach while preserving the structural properties of the image. Edge detection in image processing can be applied to the RGB images and can also be applied to the grayscale image. Applying on the grayscale images decrease the computational requirements. Several algorithms exist for edge detection, such as Prewitt, Sobel, Roberts and canny are available [10], [11], [8], out of which the performance of canny is far better. Even though it is quite old, it has become one of the standard edge detection methods and is used in our work.

The Canny Edge Detection algorithm consists of five steps:

(1) Smoothing: In this step, the image is blurred to remove noise, so that it can be used for further processing.

(2) Finding gradients: The edges should be marked where the gradients of the image has large magnitudes. It means that the edges where the intensity of the image has an abrupt change will be marked as an edge.

(3) Non-maximum suppression: Non-maximum is actually a technique to perform edge thinning. Only local maxima should be marked as edges, and edges with the values less than the local maximum are discarded to obtain more accurate and sharp edge detections.

(4) Double thresholding: Potential edges are determined by thresholding. Even after the application of non-maximum suppression, the detected edge pixels are quite good enough to present the real edge. However, there are still some edge pixels at this step caused by noise and color variation. In order to remove the noise caused by these two factors, it is required to filter out the edge pixels having the weak gradient value and only preserve the edges with the high gradient value.

(5) Edge tracking by hysteresis: After performing all the above steps we obtain almost accurate edges but still there can be edges which are separate, raising the possibility of the edges detected caused due to noise, so the final edges will be determined by suppressing all edges that are not connected to a very certain or strong edge.

B. Contour detection

Contour detection is a technique to find out the boundaries of pixels with sharp intensity changes. Contours consist of a set of points connected to each other, most likely to be located on the outlines of objects. Contour detections take a binary image as an input which is the output of the canny edge detector or a binary image obtained by applying the global thresholding technique on a grayscale image. It calculates the boundaries of objects, makes a hierarchy of the object contours to keep the record of the holes inside the parent objects. This information can be used to extract and draw any contour depending upon the user requirement.

C. Hough Transform

The Hough Transform (HT) has been recognized as a very powerful tool for the detection of parametric curves in images[12], [13]. It is implemented by a voting process that maps image edge points into manifolds in a properly defined parameter space. The Circular Hough Transform (CHT) is one of the modified versions of the HT. The purpose of using CHT is to find the circular patterns within an image scene [14]. The CHT is used to transform a set of feature points in the image space into a set of accumulated votes in a parameter space. Then, for each feature point, votes are accumulated in an accumulator array for all parameter combinations. The array elements that contain the highest number of votes indicate the presence of the shape. A circle pattern is described by

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

where x_0 and y_0 are the coordinates of the center and r is the radius of the circle.

C. Circularity

In order to detect or classify objects in images, features are extracted which are thought to provide discriminatory information. For example, to detect the presence of lobular calcifications in a digital mammogram, knowledge that such calcifications are generally spherical in shape and of higher density than surrounding tissue, motivates searching for peaks in the intensity surface with circular cross section [2]. This settles the choice of feature, but a method for extracting the feature, in this case a measure of the circularity of the region, must be selected also.

Ultimately, the feature measured is defined by the algorithm chosen for extraction, regardless of the original intention. The value of the feature in terms of discriminatory power and its accuracy in representing the intented feature are really separate issues. In most cases, however, it is highly desirable to chose methods of feature extraction which measure the intended feature faithfully.
In the case of measuring the circularity of an object, a popular technique is to measure the perimeter and the area of set of pixels representing the object and to compute.
OpenCV provides a convenient way to detect blobs and filter them based on different characteristics. There are various different parameters that control the identification process and the results.
A better and cleaner way of detecting if a geometrical shape is of circular shape using circularity

Circularity = $\frac{4\pi Area}{(perimeter)^2}$

, and decide a threshold such as 0.9, to check if the shape is circular. For perfect circle circularity == 1. You may fine tune this threshold as per your needs.

We used cv2.arcLength to find the perimeter of the contour and cv2.contourArea to get the area of the contour which are required to calculate the circularity.

RESULTS

For Good Washer:

(1) Input image

(2) Thresholding

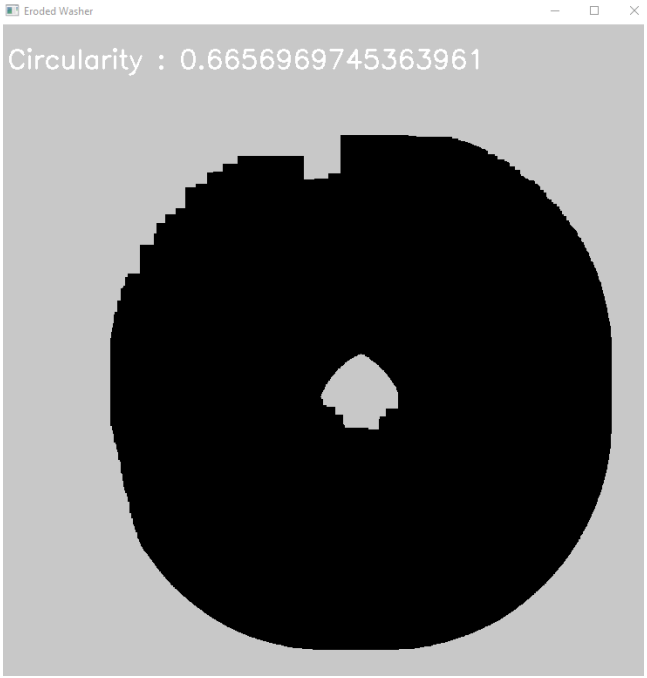
(3) Morphological Operations

(4) Detected Output

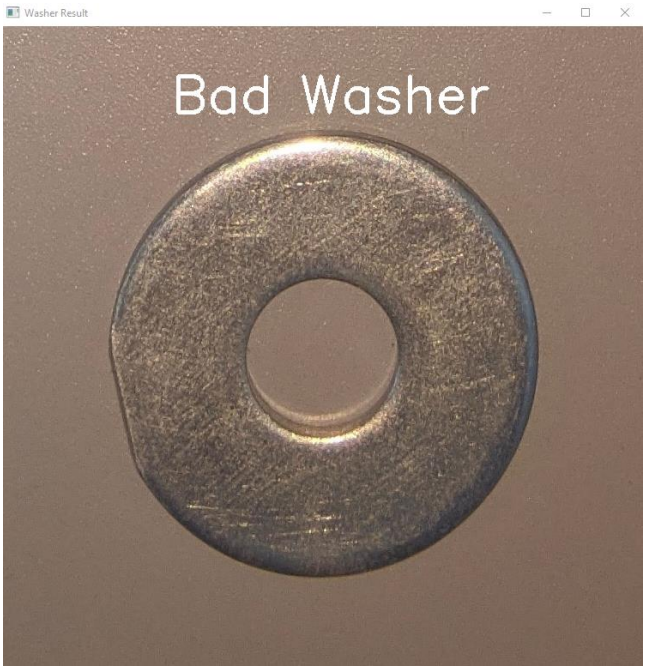
(2) Thresholding

For Defective Washer:

(1) Input image



(3) Morphological Operations



(4) Detected Output

CONCLUSION

It was found that, the perfect Washers had circularity between 0.7 & 1.0 & defective Washers had circularity between 0.4 & 0.69. Based on this value, it was possible to categorize Good & Defective Washers. This project can be used in Conveyer belts to ensure the Quality Analysis of these simple tools we call fasteners.

APPENDIX

[1] https://github.com/tanmay-redkar/washer-analysis-dip.git

[2] Application of Python-OpenCV to detect contour of shapes and colour of a real image - Mohammed Khalid Hossen - Memorial University of Newfoundland

[3] Automatic detection and counting of circular shaped overlapped objects using circular hough transform and contour detection - Jianjun Ni - Hohai University

[4] CIRCULARITY OF OBJECTS IN IMAGES - Murk J. Bottema - School of Information Science and Engineering, Flinders University of South Australia

[5] Descriptive analysis of circular data with outliers using Python programming language - N.S. Zulkipli1*, S.Z. Satari1 and W.N.S. Wan Yusoff1 - 1Centre for Mathematical Sciences, College of Computing and Applied Sciences, Universiti Malaysia Pahang

[6] Fast algorithm for Multiple-Circle detection on images using Learning Automata - Erik Cuevasa, Fernando Warioa, Valentín Osuna-Encisob, Daniel Zaldivara and Marco Pérez-Cisnerosa - Departamento de Ciencias Computacionales, Universidad de Guadalajara, CUCEI