
Box Detection with OpenCV

Alperen Bitirgen

1 Retrieving Files From Given Directory

First, I saved the image file names in a given directory into a vector object. Then I read the image from the directory until the batch-size and saved it in a local variable. At each step, I processed up to 1 batch-size image and then saved the results.

2 Processing Images

2.1 Color/Intensity Adjustments

I started my process by converting the pictures I read into grayscale form. Then I applied Gamma correction, because most of the pictures were in very dark tones and a non-linear intensity shift operation to make the background-object separation will help us in the next steps.

2.2 Edge Detection

After completing the basic color and intensity adjustment, I decided to do edge detection with the help of Canny edge detection. After this step, as I guessed, the perforated parts on the objects caused us to get a very noisy result. So after Canny, I tried to eliminate these noisy parts with the help of Gaussian blur.

2.3 Contours and Min Area Rectangles

After Nois removal, I used the findContours function in OpenCV to find the contour. Right after that, I detected the rectangles with the minimum area over these contours. However, since it was not possible to completely eliminate the noise in the result I obtained in the previous step, the rectangles I obtained were also very noisy (very small dimensions, rectangles interlocked at different angles, etc.). I decided to create a filtering method consisting of a 2 steps on these rectangles.

2.4 Filtering the Minimum Area Rectangles

First I filtered the rectangles based on their area and position. Since some of the objects in the rectangles close to the edges and corners may not be visible in the picture, they can form rectangles with smaller areas compared to the ones closer to the center.

In the second step, I examined the overlap of the rectangles whose centers are close to each other, and I chose one of the rectangles that have a center at that periphery, so that it has a larger scope. Before starting this step, I created a filter that assumed all rectangles would be preserved, and then eliminated some of them as I compared the rectangles.

2.5 Writing Some of the Details on the Image

In the last part about the processes; First I drew the rectangles, the corner points and coordinates of the rectangles. Then I placed the angle, area, height and width information of the rectangles near the center of the area covered by the rectangle.

3 Saving Processed Images

I went to the directory where the images were taken by the user and overwritten the original image of the image I obtained as a result of the operations.