

Count non-zero pixels in area rotated rectangle

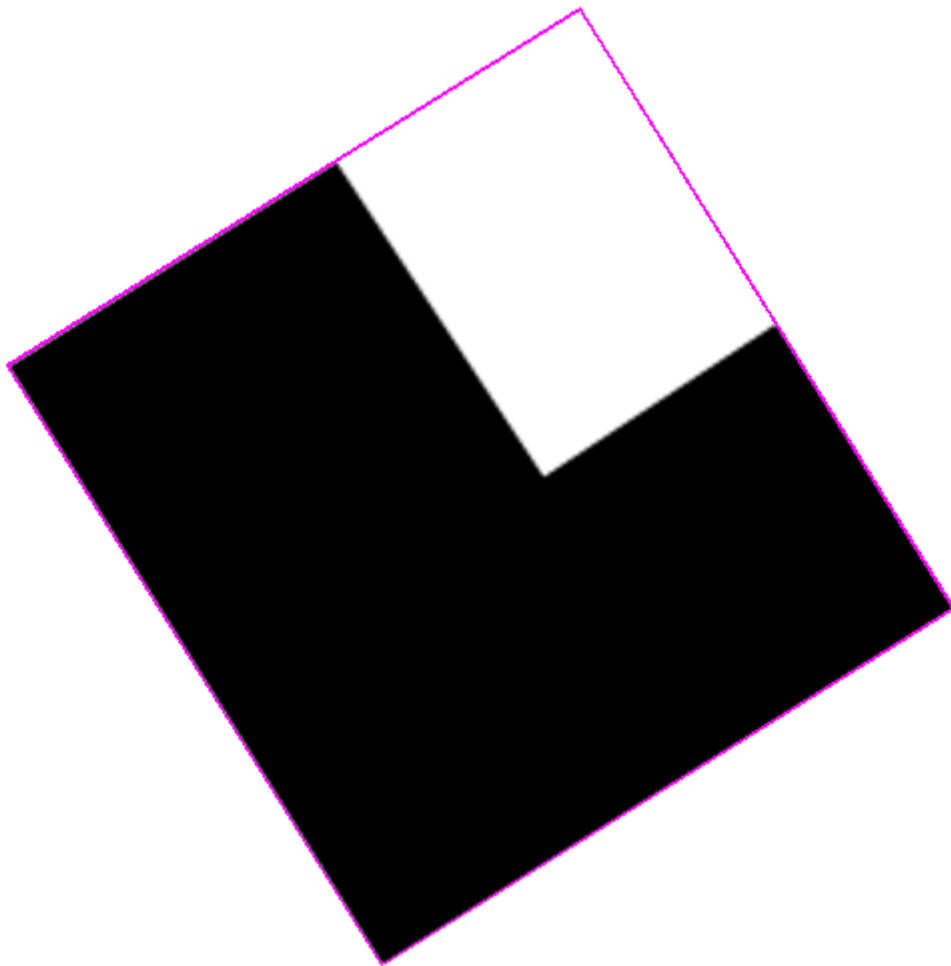
Asked 2 years, 9 months ago Active 2 years, 6 months ago Viewed 757 times



1



I've got a binary image with an object and a rotated rectangle over it, found with `cv2.findContours` and `cv2.minAreaRect`. The image is normalized to `[0;1]` What is the most efficient way to count non-zero area within the bounding rectangle?



[python-3.x](#) [opencv](#) [computer-vision](#)

asked Nov 11 '17 at 15:28



[Sasha Korekov](#)

173 1 15



if you need even more performance, adapt the `drawContours` code to count the pixels instead of drawing. – [Micka](#) Nov 11 '17 at 17:07

3 Answers

Active	Oldest	Votes
--------	--------	-------



2



- Create new zero values Mat that has the same size of your original image.
- Draw your rotated rectangle on it in (fillConvexPoly using the RotatedRect vertices).
- Bitwise_and this image with your original mask
- apply findnonzero function on the result image



You may also apply the previous steps on ROI of the image since you have the bounding box of your rotated rectangle.



answered Nov 11 '17 at 15:32

[Humam Helfawi](#)

17k 12 54 123



2



According to Humam Helfawi's answer I've tuned a bit suggested steps, so the following code seems doing what i need:

```
rectangles = [(cv2.minAreaRect(cnt)) for cnt in contours]
for rect in rectangles:
    rect = cv2.boxPoints(rect)
    rect = np.int0(rect)
    coords = cv2.boundingRect(rect)
    rect[:,0] = rect[:,0] - coords[0]
    rect[:,1] = rect[:,1] - coords[1]
    area = cv2.contourArea(rect)
    zeros = np.zeros((coords[3], coords[2]), np.uint8)
    cv2.fillConvexPoly(zeros, rect, 255)
    im = greyscale[coords[1]:coords[1]+coords[3],
    coords[0]:coords[0]+coords[2]]
    print(np.sum(cv2.bitwise_and(zeros, im))/255)
```

answered Nov 11 '17 at 17:01

[Sasha Korekov](#)

173 1 15



1



contours is a list of points. You can fill this shape on an empty binary image with the same size using cv2.fillConvexPoly and then use cv2.countNonZero or numpy.count_nonzero to get the number of occupied pixels.

answered Feb 16 '18 at 15:01

[Ozan Akyıldız](#)

15 5