PYIMAGESEARCH

IMAGE PROCESSING (HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/IMAGE-PROCESSING/)

TUTORIALS (HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/TUTORIALS/)

Detecting Circles in Images using OpenCV and Hough Circles

Figure 2: Detecting the top of a soda can using circle detection with OpenCV.

by Adrian Rosebrock (https://www.pyimagesearch.com/author/adrian/) *on* July 21, 2014

(https://pyimagesearch.com/wp-content/uploads/2014/07/detect_circles_soda.jpg)

A few days ago, I got an email from a PyImageSearch reader asking about circle detection. See below for the gist:

> *Hey Adrian*

> *(https://pyimagesearch.com/2014/04/21/building-pokedex-python-finding-game-boy-screen-step-4-6/) in images, but I was wondering, how do you detect circles in images using OpenCV?*
>
> *Thanks.*

Great question.

As you've probably already found out, detecting circles in images using OpenCV is substantially harder than detecting other shapes with sharp edges.

But don't worry!

In this blog post I'll show you how to utilize the `cv2.HoughCircles` function to detect circles in images using OpenCV.

To learn how to detect circles with OpenCV, *just keep reading!*

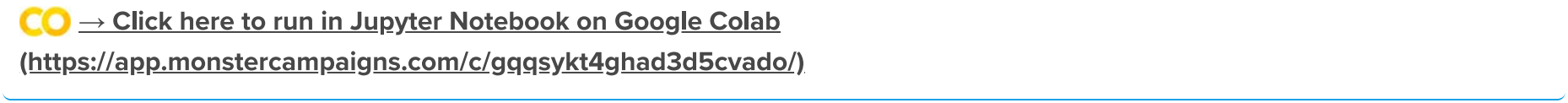
Looking for the source code to this post?

JUMP RIGHT TO THE DOWNLOADS SECTION →

The cv2.HoughCircles Function

In order to detect circles in images, you'll need to make use of the `cv2.HoughCircles` function. It's definitely not the easiest function to use, but with a little explanation, I think you'll get the hang of it.

Take a look at the function signature below:




```
Detecting Circles in Images using OpenCV
cv2.HoughCircles(image, method, dp, minDist)
```

- **`image`:** 8-bit, single channel image. If working with a color image, convert to grayscale first.
- **`method`:** Defines the method to detect circles in images. Currently, the only implemented method is

- **dp:** This parameter is the inverse ratio of the accumulator resolution to the image resolution (see Yuen et al. for more details). Essentially, the larger the dp gets, the smaller the accumulator array gets.
- **minDist:** Minimum distance between the center *(x, y)* coordinates of detected circles. If the `minDist` is too small, multiple circles in the same neighborhood as the original may be (falsely) detected. If the `minDist` is too large, then some circles may not be detected at all.
- **param1:** Gradient value used to handle edge detection in the Yuen et al. method.
- **param2:** Accumulator threshold value for the `cv2.HOUGH_GRADIENT` method. The smaller the threshold is, the more circles will be detected (including false circles). The larger the threshold is, the more circles will potentially be returned.
- **minRadius:** Minimum size of the radius (in pixels).
- **maxRadius:** Maximum size of the radius (in pixels).

If this method seems complicated, don't worry. It's actually not too bad.

But I will say this — be ready to play around with the parameter values from image to image. The `minDist` parameter is especially important to get right. Without an optimal `minDist` value, you may end up missing out on some circles, or you may detecting many false circles.

Detecting Circles in Images using OpenCV

Ready to apply the `cv2.HoughCircles` function to detect circles in images?

Great. Let's jump into some code:


```
Detecting Circles in Images using OpenCV
1. # import the necessary packages
2. import numpy as np
3. import argparse
4. import cv2
5.
6. # construct the argument parser and parse the arguments
7. ap = argparse.ArgumentParser()
8. ap.add_argument("-i", "--image", required = True, help = "Path to the image")
9. args = vars(ap.parse_args())
```

Lines 2-4 import the necessary packages we'll need. We'll utilize NumPy for numerical processing, `argparse` for parsing command line arguments, and `cv2` for our OpenCV bindings.

Then, on **Lines 7-9** we parse our command line arguments. We'll need only a single switch, `--image`, which is the path to the image we want to detect circles in.

Let's go ahead and load the image:

```
11. | # load the image, clone it for output, and then convert it to grayscale
12. | image = cv2.imread(args["image"])
13. | output = image.copy()
14. | gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

We load our image off disk on **Line 12** and create a copy of it on **Line 13** so we can draw our detected circles without destroying the original image.

As we'll see, the `cv2.HoughCircles` function requires an 8-bit, single channel image, so we'll go ahead and convert from the RGB color space to grayscale on **Line 14**.

Okay, time to detect the circles:


```
Detecting Circles in Images using OpenCV
16. | # detect circles in the image
17. | circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1.2, 100)
18. |
19. | # ensure at least some circles were found
20. | if circles is not None:
21. |    # convert the (x, y) coordinates and radius of the circles to integers
22. |    circles = np.round(circles[0, :]).astype("int")
23. |
24. |    # loop over the (x, y) coordinates and radius of the circles
25. |    for (x, y, r) in circles:
26. |        # draw the circle in the output image, then draw a rectangle
27. |        # corresponding to the center of the circle
28. |        cv2.circle(output, (x, y), r, (0, 255, 0), 4)
29. |        cv2.rectangle(output, (x - 5, y - 5), (x + 5, y + 5), (0, 128, 255), -1)
```

Detecting the circles is handled by the `cv2.HoughCircles` function on **Line 17**. We pass in the image we want to detect circles as the first argument, the circle detection method as the second argument (currently, the `cv2.cv.HOUGH_GRADIENT` method is the only circle detection method supported by OpenCV and will likely be the only method for some time), an accumulator value of 1.5 as the third argument, and finally a `minDist` of 100 pixels.

A check is made on **Line 20** to ensure at least one circle was found in the image.

Line 22 then handles converting our circles from floating point *(x, y)* coordinates to integers, allowing us to draw them on our output image.

From there, we start looping over the center *(x, y)* coordinates and the radius of the circle on **Line 25**.

We draw the actual detected circle on **Line 28** using the `cv2.circle` function, followed by drawing a rectangle at the center of the circle on **Line 29**.

Finally, **Lines 32 and 33** display our output image.

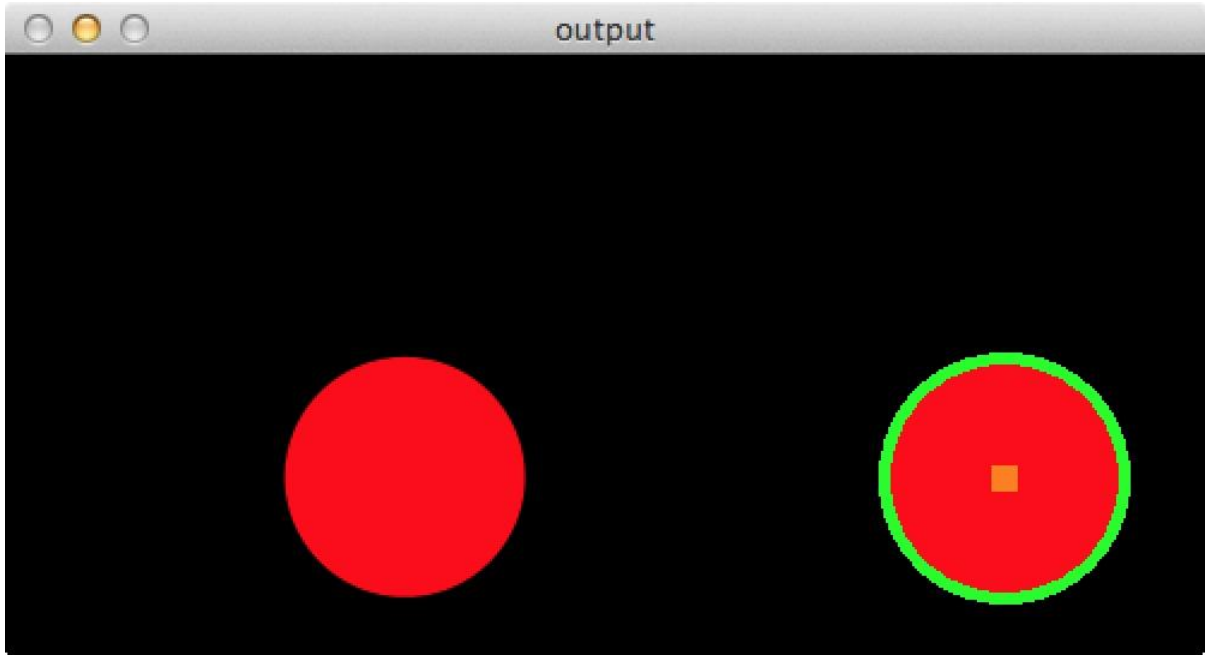
So there you have it — detecting circles in images using OpenCV.

But let's go ahead and take a look at some results.

Fire up a shell, and execute the following command:

```
1. | $ python detect_circles.py --image images/simple.png
```

We'll start with something simple, detecting a red circle on a black background:



(https://pyimagesearch.com/wp-content/uploads/2014/07/detect_circles_simple.jpg)

Figure 1: Detecting a simple circle in an image using OpenCV.

Not bad! Our Python script has detected the red circle, outlined it in green, and then placed an orange square at the center of it.

Let's move on to something else:

```
Detecting Circles in Images using OpenCV
$ python detect_circles.py --image images/soda.png
```

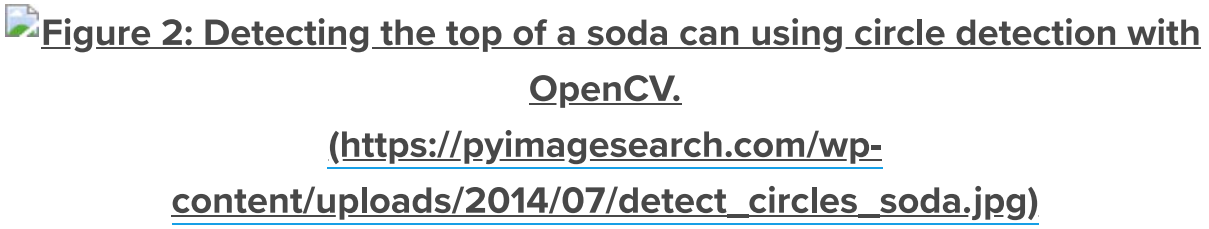


Figure 2: Detecting the top of a soda can using circle detection with OpenCV. (https://pyimagesearch.com/wp-content/uploads/2014/07/detect_circles_soda.jpg)

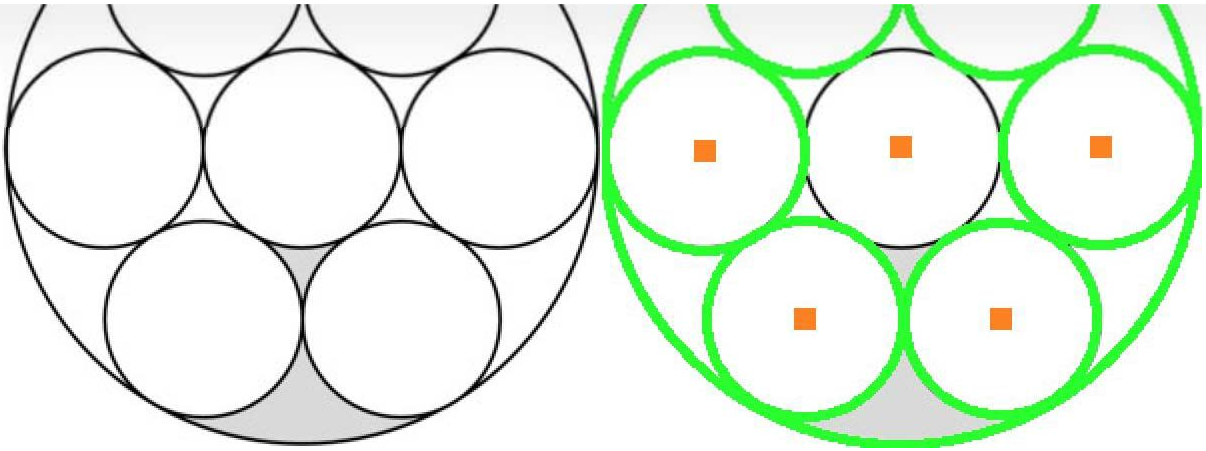
Figure 2: Detecting the top of a soda can using circle detection with OpenCV.

Again, our Python script is able to detect the circular region of the can.

Now, let's try the 8 circle problem.

In this problem we have *one* large circle, followed by *seven* circles placed inside the large one.

Since this is a much smaller image than the previous ones (and we are detecting multiple circles), I'm going to adjust the `minDist` to be 75 pixels rather than 100.



(https://pyimagesearch.com/wp-content/uploads/2014/07/detect_circles_8circles.jpg)

Figure 3: Notice how `cv2.HoughCircles` failed to detect the inner-most circle.

Hm. Now it looks like we have ran into a problem.

The `cv2.HoughCircles` function was able to detect only *seven* of the circles instead of all *eight*, leaving out the one in the center.

Why did this happen?

It's due to the `minDist` parameter. The center *(x, y)* coordinates for the large outer circle are identical to the center inner circle, thus the center inner circle is discarded.

Unfortunately, there is not a way around this problem unless we make `minDist` unreasonably small, and thus

Summary

In this blog post I showed you how to use the `cv2.HoughCircles` function in OpenCV to detect circles in images.

Unlike detecting squares or rectangles in images, detecting circles is substantially harder since we cannot reply on approximating the number of points in a contour.

To help us detect circles in images, OpenCV has supplied the `cv2.HoughCircles` function.

While the `cv2.HoughCircles` method may seem complicated at first, I would argue that the most important parameter to play with is the `minDist`, or the minimum distance between the center *(x, y)* coordinates of detected circles.

If you set `minDist` too small, you may end up with many falsely detected circles. On the other hand, if `minDist` is too large, then you may end up missing some circles. Setting this parameter definitely takes some fine tuning.

Have a Question?

Do you have a question about OpenCV and Python? Just send me a message. And I'll do my best to answer it on this blog.

Download the Source Code and FREE 17-page Resource Guide

Enter your email address below to get a .zip of the code and a **FREE 17-page Resource Guide on Computer Vision, OpenCV, and Deep Learning.** Inside you'll find my hand-picked tutorials, books, courses, and libraries to help you master CV and DL!

The *OCR with OpenCV, Tesseract, and Python* IndieGoGo campaign is *LIVE!*

Get 25-35% OFF my books and courses (including my brand new OCR book). (https://www.indiegogo.com/projects/ocr-with-opencv-tesseract-and-python/coming_soon/x/24003505)

About the Author

Hi there, I'm Adrian Rosebrock, PhD. All too often I see developers, students, and researchers wasting their time, studying the wrong things, and generally struggling to get started with Computer Vision, Deep Learning, and OpenCV. I created this website to show you what I believe is the best possible way to get your start.

Previous Article:

(https://www.pyimagesearch.com/2014/07/14/3-ways-compare-histograms-using-opencv-python/)

Next Article:

Segmentation: A SLIC Superpixel Tutorial using Python

(https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/)

192 responses to: Detecting Circles in Images using OpenCV and Hough Circles

Nick R

January 12, 2015 at 4:51 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-266737)

Hi! Great tutorial, I just had a quick question – if I am doing this on Windows and not Linux how would I go about doing the same thing? Or would I have to run this through a windows command prompt?

Adrian Rosebrock

Hi Nick, you would have to run the script from the Windows command prompt.

ATUL

September 1, 2017 at 2:44 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-433821)

Hey.. Adrian.
Can you provide me a code for this picture. I want to find out the Coordinates of holes in the given picture.
Please Help me
.
Thanx in advance.

Adrian Rosebrock

September 1, 2017 at 9:41 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-433839)

While I'm happy to help, I cannot write code for you. I hope you understand.

cf

September 5, 2017 at 2:01 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-

Do you also use c++ programming language to describe algorithm

Adrian Rosebrock

September 5, 2017 at 9:12 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-434120)

I only provide Python code here on PyImageSearch.

Dustin Decker

January 26, 2015 at 3:57 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-272969)

FINALLY! I'm guessing my search foo was lacking, but this is the first place I've found a concise walk through of the cv2.HOUGH_GRADIENT method, particularly the MinDist impact... for the past 24 hours I've been trying random values to deduce how all this was working.

THANK YOU!

Hi Dustin. Awesome! I'm glad to hear that the tutorial was helpful for you!

Ifeoma

January 28, 2015 at 10:51 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-274037)

Thanks so much for the detailed explanation. I used it to detect circles in my Android application when pictures are taken with the camera. It worked fine and saved me a lot of time. Thanks.

Adrian Rosebrock

January 29, 2015 at 6:49 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-274141)

Fantastic, I'm happy the article helped!

What a great time to learn python image recognition. Thanks.

Sofia

February 21, 2015 at 8:34 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-280838)

Hey Adrian!

Thanks for the run-through.

Is there an intuitive way to understand the meaning of param1, param2, and dp?

Moreover, I am confused because: my images have many circle-like objects, but if I increase the range of radii by increasing maxRadius, I sometimes get fewer circles. Is this something you have seen and could explain?

Thanks a lot!

Hi Sofia, if you have many circle like objects and you increase the maxRadius, then you will certainly find fewer circles. The maxRadius parameter controls the maximum radius of a circle. So if a circle has a radius greater than maxRadius, then it will not be detected. I would also take a look at **these slides (https://www.cis.rit.edu/class/simg782/lectures/lecture_10/lec782_05_10.pdf)** for a more in-depth review of circle detection.

Jayanthi.P

March 6, 2015 at 6:04 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-284196)

I run this code but i got an error in line no 22.It displayed error like IndentationError:expected an indented block.How to run this code without an error.Can you help me sir?.Thank you.

Adrian Rosebrock

March 6, 2015 at 6:28 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-284208)

Based on that error it looks like your Python code is not indented correctly — perhaps you mixed both spaces

Jayanthi.P

March 7, 2015 at 3:34 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-284613)

Sir what do you mean python code is not indented correctly?.In line no 22 you use astype(“int”) that int show in error.If i change int like float,i got the same error.What i suppose to do?.

Adrian Rosebrock

March 7, 2015 at 6:50 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-284668)

The “int” or “float” is not the issue. The problem is you may have mixed tabs and spaces in your indentation of the code. More information on this type of error can be found in **this (http://stackoverflow.com/questions/10238770/indentationerror-expected-an-indented-block)** StackOverflow post. If you can spare the time, brushing up on a bit of Python will definitely help you avoid these types of errors.

Rama Chandran.P

Sir i am doing my project on human computer interaction.In that i plan to do roak paper sscissor game and arithmetic operation.So i have an hand image which capture from webcam.From that captured image i have to fix center point and contour point.So that only i can detect fingers.Sir can you tell me how to fix these points and how to detect finger from the hand image?.Thank you!

Adrian Rosebrock

March 9, 2015 at 8:27 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-285472)

Hi Rama, you'll have to look into the topic of "contour defects" to aide in detecting fingers connected to the hand.

Vinh

March 17, 2015 at 11:03 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-290130)

Hello

It is a great tutorial.

(something like: **https://mysticalbootcamp.files.wordpress.com/2011/12/3.jpg (https://mysticalbootcamp.files.wordpress.com/2011/12/3.jpg)**)

Now I want to measure the radius of this kind of “circle”. How can I start?

Thank you very much

Adrian Rosebrock

March 17, 2015 at 11:33 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-290149)

There are different ways to approach a problem like this. You could try more advanced techniques of ellipse detection, those would probably help. Personally, I would just find the contours of each circle, compute the center of the circle, and from there, it’s simple to determine the radius.

Mon

April 8, 2015 at 11:40 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-303161)

Adrian Rosebrock

April 8, 2015 at 11:42 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-303162)

Line 25 gives you the x, y coordinates of the center of the circle along with the radius.

Heyne

April 9, 2015 at 12:03 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-303526)

Can I detect circles in video by picamera and how? Thanks so much

Adrian Rosebrock

April 9, 2015 at 12:57 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-303544)

You certainly can, but you'll have to tune the parameters of the `cv2.HoughCircles` function, which is not always

Heyne

April 13, 2015 at 9:38 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-304324)

Thanks, I've tried but have not yet done. what should I do next?

Adrian Rosebrock

April 13, 2015 at 11:32 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-304333)

The **scikit-image (http://scikit-image.org/docs/dev/auto_examples/plot_circular_elliptical_hough_transform.html)** package as a more powerful transform that would probably be worth looking into.

Steve Carter

December 10, 2016 at 9:39 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-413163)

Hi Adrian,

Is that an easier task than simply detecting circles from a raw stream?
Thanks
Steve

Adrian Rosebrock
December 12, 2016 at 10:44 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-413312)

A video stream is just a collection of frames. Each frame can be considered an individual image. A video stream can be slightly more complicated due to motion blur as objects move, but the same general process applies. Technically applying circle detection to a single image (provided there is no blur) is easier, but if you can guarantee that in a video stream, it's just as easy.

Adnan
March 11, 2019 at 9:44 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-506076)

Hi Adrian,

When I try to detect cirlcles drawn on a paper, it doesn't work, It can detect circular objects(coins, cans)

How can I solve it?

Adrian Rosebrock

March 13, 2019 at 3:35 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-506549)

Depending on the input images the circle may be distorted or cannot be segmented. It's impossible to know without seeing your example images. You may want to try training your own custom object detector. **HOG + Linear SVM (https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/)** would be a good first step.

Dave

April 27, 2015 at 11:58 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-309728)

Hey!! did you solve it?

I need to do exactly the same, and i don't know how to do that. If you both please can help me, i'll be very grateful.

I'm using a picamera with a raspberry pi B+.

Elmo

June 28, 2015 at 1:21 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-331241)

Hello adrian, is it possible to detect an oval object as a circle too? i mean the object is not a perfect circle. i've implemented your tutorial but it cannot detect the non-perfect-circle object. or maybe which params that i need to change? thanks a lot before

---

Adrian Rosebrock

June 28, 2015 at 2:37 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-331272)

You can indeed detect an ellipse/oval region instead of a circle in an image, but it's a bit more challenging (especially for the algorithm itself). Take a look at the circular and elliptical Hough transforms of **scikit-image (http://scikit-image.org/docs/dev/auto_examples/plot_circular_elliptical_hough_transform.html)** for more information.

September 10, 2015 at 1:28 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-356578)

Hi Adrian,

You are doing a fantastic job. Your blog has now become my preferred destination to search for any python + opencv feature. I was wondering if you plan to write an article on multiprocessing of images for increasing speed anytime soon.
So I want to batch process some images from a given folder and save the output in another folder. I tried using Pool from multiprocessing library but am running into errors. Any pointers?

Adrian Rosebrock

September 11, 2015 at 6:31 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-356949)

Hey Pranav — thanks for putting this back on my radar. I was planning on doing a series of posts on Hadoop and image processing in the future, but I should start with just the basics of multiprocessing. As for other libraries, you might want to give **pp (http://www.parallelpython.com/)** a try.

Hi, Cool stuff! I'm trying to make it detect my iris but no luck, even if I open my eyes really wide... Any suggestions?

Adrian Rosebrock

September 14, 2015 at 6:15 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-358375)

If you're trying to detect your iris, then I'm not sure circle detection is your best bet. I would try simple thresholding methods (since there will be substantial contrast between your iris and the whites of your eyes) first.

Ibrahim

October 3, 2015 at 5:27 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-365729)

Hi Adrian,

Thanks for the tutorials. I'm having trouble with detecting circles using OpenCV 3 with Python3. I can load and

'module' object has no attribute 'cv'.

Changing this to cv2.CV_HOUGH_GRADIENT I get a different error

'module' object has no attribute 'CV_HOUGH_GRADIENT'.

I am working within the virtual environment (if that's the correct terminology).

Any help would be appreciated.

Cheers,
Ibrahim

Adrian Rosebrock
October 3, 2015 at 6:15 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-365751)

Again, just to firm: you're using OpenCV 3? If so, then it should be: `cv2.HOUGH_GRADIENT`

I am using the Python/OpenCV that I created using your:

https://www.pyimagesearch.com/2015/12/14/installing-opencv-on-your-raspberry-pi-zero/ (https://www.pyimagesearch.com/2015/12/14/installing-opencv-on-your-raspberry-pi-zero/)

When I run your code here, I get:

(cv) pi@raspberrypi:~ $ python detect_circles.py –image Hough_image1.jpg
...
circles = cv2.HoughCircles(gray, cv2.CV_HOUGH_GRADIENT, 1.2, 100)
AttributeError: 'module' object has no attribute 'CV_HOUGH_GRADIENT'
...

Not sure how to fix this one yet.

Adrian Rosebrock

May 25, 2017 at 4:25 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-425964)

Hi John — it sounds like you are using OpenCV 3 while this blog post assumes you're using OpenCV 2.4. In

Mohamed O.

August 29, 2017 at 1:59 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-433598)

Thanks. I faced the same issue & changing it to cv2.HOUGH_GRADIENT worked for me.

Talat Shaikh

October 20, 2015 at 12:13 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-373271)

Hi Adrian,

How do I output the number of circles detected on the terminal screen?\
Thanks.

Adrian Rosebrock

October 20, 2015 at 6:13 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-373351)

The `circles` variable is just a list, so just use the `len` function:

akif

November 25, 2015 at 11:10 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-380747)

Hi there,
How can we find the radius of circle?

Adrian Rosebrock

November 25, 2015 at 1:52 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-380768)

Hey Akif — you'll want to look into the `cv2.minEnclosingCircle` function. This will give you the radius of the circle.

akif

November 27, 2015 at 4:12 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-381111)

I am trying to do iris recognition system.For the last step,I need to encode the normalized image into binary and then,match the encoded images.
For encoding,I used cv2.imencode,is this right thing to do?
For matching,I could not find what to do.
Thank you...

Adrian Rosebrock

November 28, 2015 at 6:50 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-381186)

If you are trying to compare images for similarity/match them, you should look into **template matching (https://www.pyimagesearch.com/2014/01/27/hobbits-and-histograms-a-how-to-guide-to-building-your-first-image-search-engine-in-python/)**, **MSE/SSIM (https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/)**, or even using a **distance metric ()**.

Ty

December 7, 2015 at 3:18 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-382423)

Thanks for the tutorials Adrian

I can detect circles when I use the sample images you provided but when I try to read other images like this image **http://www.clker.com/cliparts/U/D/6/q/l/q/12-color-circles-hi.png (http://www.clker.com/cliparts/U/D/6/q/l/q/12-color-circles-hi.png)** , the hough circles return None as the value meaning no circles were detected. The parameters I used were 1.2 as dp and 75 as minDist. Am I doing something wrong?

Thanks in advance

Hi again,

I think I found a solution. It had something to do with the image size. After resizing the image to 300×300, I was able to detect some circles.

I have another image with dimension 640×480. I tried resizing it to 300×300 but I believe the circles in the original

Thanks in advance

Adrian Rosebrock

December 7, 2015 at 6:56 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-382449)

The images do not have to be a perfect square, they can be rectangular, that's not an issue at all. Often times it's beneficial to resize your images and make them smaller before you process them.

Mike

January 8, 2016 at 9:49 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386097)

Hi Adrian, I'm prefacing this with: I am very new to python or opencv

is there some way to name or number the circles I detect? Not just print the total number found, but place a unique

For instance, if I have an image with 5 circles, can I place a "1" on the first one, a "2" on the second, etc.
Thanks

Adrian Rosebrock
January 9, 2016 at 6:19 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386132)

Since `circles` is just a list, I would assign a unique identifier to each one pasted on the index of the circle in the list. For example

```
for (i, (x, y, r)) in enumerate(circles):
```

Will assign the unique index `i` to each circle in the `circles` list.

Selim
January 9, 2016 at 9:18 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386143)

Hi,

there a way to solve this problem?

I recently started learning raspberry, so an easy explanation is much appreciated.

Thanks

Adrian Rosebrock

January 9, 2016 at 5:34 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386168)

Hough circles is not a good method for real-time video processing. The motion blur makes it very challenging to reliably detect the circles. If possible, you might want to try **color based methods (https://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/)** or using **HOG + Linear SVM (https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/)**.

moti

January 14, 2016 at 4:38 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386777)

moti

January 14, 2016 at 4:46 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386783)

i will add some agenda,
1. i am following a ball with this method and point it with laser and servos, to make the “hit” more accurate i need good detection (parameters 1 &2).

2. i try to run it in realtime, the loop time is ~180 ms with resolution of 640X480.
to reduce the loop time i am cropping the image according to last recognition,
thats make the loop time ~40 ms.
BUT when running the SAME picture – the biggest picture find the circle just fine and with the cropped image it sometimes find bigger radius detection (with the same parameters).
do you have an idea why the processing is different???
thanks again

Adrian Rosebrock

January 14, 2016 at 6:12 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386796)

(https://www.pyimagesearch.com/2016/01/04/unifying-picamera-and-cv2-videocapture-into-a-single-class-with-opencv/), as well as the posts it links to. Inside I detail how to speedup camera I/O substantially — this will help with your first problem.

As for your second question, you're running Hough circles on a smaller, cropped ROI? This actually does make sense due to the accumulator gradient. See the paper referenced in the blog post for more information.

Adrian Rosebrock
January 14, 2016 at 6:09 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386795)

For a deeper explanation of the parameters, review the **Yuen et al. paper (http://www.bmva.org/bmvc/1989/avc-89-029.pdf)**. This paper discusses the gradient value along with the accumulator.

moti
January 14, 2016 at 8:40 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-386817)

thanks.

January 26, 2016 at 10:56 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-388145)

Thank you for this helpful tutorial!
I am using this code you have provided to detect petri dishes for a project my lab is working on. I am having a few problems where the code is detecting a circle but it is off center from what I expect and doesnt match the contours of the petri dish. It also detects some circles that dont exist even though my min distance is 1000, which I think is quite high.
I am new to image analysis and opencv and generally dont have much clue as to what I am doing, so I would really appreciate some help!

Adrian Rosebrock

January 27, 2016 at 3:08 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-388219)

It's hard to say what the exact issue could be without seeing your images, but in general, it can be hard to determine the parameters to Hough circles. You might want to investigate simple contour methods instead using the `cv2.findContours` function and contour properties to identify circle-like regions. You can also try the **scikit-image implementation for finding circles. (http://scikit-image.org/docs/dev/auto_examples/edges/plot_circular_elliptical_hough_transform.html)**

YaddyVirus

March 27, 2016 at 11:09 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-393044)

Hi Adrain! I tried running your code but it gives me an error saying-

output = image.copy()
AttributeError: 'NoneType' object has no attribute 'copy'

I have removed your input argument function and provided a direct image to the cv2.imwrite function. Could that be the problem? Because it looks like it is an input error.

Adrian Rosebrock

March 28, 2016 at 1:34 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-393177)

If you hardcoded your image path to `cv2.imread` and your image is `None`, then you'll want to double check the path you supplied to `cv2.imread` and ensure that it's correct. Based on the error message, I can almost guarantee that the path to the image is not valid.

March 31, 2016 at 7:10 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-393553)

Hi,after detecting circles with houghcircle ,i need to verify if all circles (wanted to be detected) is detected
as a first solution is to add reference image which contained all circles must be detected
and try to color the circle detected in the reference image
i asked if it's possible to do this with opencv.If,yes i need some keywords or helpful tutoriel
If,no there is other solution and thanks to reply

Adrian Rosebrock

April 1, 2016 at 3:17 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-393638)

Hey Lali — I'm not sure I understand your question. What do you mean by "verify if all circles are detected"?

Palanikumar

April 2, 2016 at 3:18 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-393690)

Hey Adrian.. thanks for the tutorial. I'm having problem with executing the code. i encountered on 20th line

i tried with the change of cv to cv2 since i'm using opencv3..
kindly give solution to this problem.. Thanks in advance

Adrian Rosebrock

April 3, 2016 at 10:30 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-393811)

Indeed, this is an error related to OpenCV 3 (this blog post was originally written for OpenCV 2.4). You can resolve the error by changing `cv2.cv.CV_HOUGH_GRADIENT` to `cv2.HOUGH_GRADIENT`.

bo (http://bobofuk95@gmail.com)

April 4, 2016 at 6:31 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-393925)

thanks for your tutorial
i have a question :if i want to get the coordinate of circle center，what should i do?

Take a look at **Line 25**. The x and y values contain the center of the circle.

Harshita Mangal

May 18, 2016 at 4:41 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-397995)

Hi, when I try to run this code I am getting an error: module ‘cv2’ has no attribute ‘cv’. I have openCV installed. Also, I tried the suggestion on stackoverflow to write import cv2.cv as cv and then use cv instead of cv2.cv but it still doesn’t work. Your help would be highly appreciated. Thanks in advance.

Adrian Rosebrock

May 19, 2016 at 6:07 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-398199)

It seems like you’re using OpenCV 3 — this tutorial was designed for OpenCV 2.4.X. Just change `cv2.cv.CV_HOUGH_GRADIENT` to `cv2.HOUGH_GRADIENT` and it will work.

July 23, 2016 at 2:26 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-402864)

I would like to detect a circle in an image and after that, would have to convert the circle into a rectangular image using a polar conversion tool for rectangular . I'm working on C ++ , Visual Studio 2012, opencv 2.4.13 .
The circle I'm managing to find the image but not can do the conversion , someone would have an idea of how to do this ? Thank you very much in advance.

Sorry about my English.

vinod

August 3, 2016 at 7:52 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-403475)

hello,

i have a question, how to detect these circles with driod camera,

---

I haven't used a Droid camera before, but if you can access the camera via `cv2.VideoCapture`, you'll be able to read frames from the camera and process them individually for circles.

vinod

August 8, 2016 at 3:02 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-403854)

thank you so much Adrian , i developed the code , with help of your code, one more help i need how to calculate distance from camera to circle(object) .

Adrian Rosebrock

August 8, 2016 at 6:37 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-403870)

Computing the distance from a camera to an object **is covered in this tutorial**

Brian

September 5, 2016 at 9:06 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-405581)

Where are Param1, Param2, MinRadius, and MaxRadius used in your code? Thanks!

Adrian Rosebrock

September 5, 2016 at 12:47 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-405597)

Those parameters are optionally/implicitly supplied to `cv2.HoughCircles` on Line 17. I used the default values.

LinuxCircle (http://www.linuxcircle.com)

September 13, 2016 at 8:17 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-406120)

Your articles have been helping me teach in Australia via LinuxCircle.com

1. How do we increase the chance of detecting just the one ball?
2. Will image smoothing such as blurring method help in segregating the ball with the background?
3. What is the role of light in Hough method? If the robot light a LED torch towards the ball will it increase the chance of being detected?

Adrian Rosebrock

September 15, 2016 at 9:36 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-406181)

In general, I wouldn't recommend Hough circles for this. The parameters are tough to get right, especially in a real-time setting. Instead, I would use **something like this tutorial (https://www.pyimagesearch.com/2015/09/21/opencv-track-object-movement/)** for ball tracking.

Bv

September 24, 2016 at 6:26 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-406647)

Such a nice tutorial,So what is the use circle detection in computer vision and navigation, how important is this.

Adrian Rosebrock

September 27, 2016 at 8:50 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-406820)

Can you elaborate on what you mean by “navigation”? What are you trying to accomplish?

irving

October 3, 2016 at 11:38 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-407445)

where should I put the path of my image?

Adrian Rosebrock

October 4, 2016 at 6:52 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-407471)

You can put the image wherever you like it — you just need to supply the path to the image via command line

satria

November 17, 2016 at 4:55 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-410969)

hi adrian...
i have some problem
when i run the program

output = image.copy()
AttributeError: 'NoneType' object has no attribute 'copy'

what happend???

Adrian Rosebrock

November 18, 2016 at 8:55 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-411033)

If you are getting an error related to "NoneType" right after `cv2.imread` is being called, then 99% of the time is

GK_Bats

December 8, 2016 at 4:54 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-412994)

Hi Adrian,
These are awesome tutorials.
I'm trying to detect droplets in an image. But to start with, I tried executing your code, it worked fairly well for your sample images. But, for my images, it did not work. The cmd simply skipped to the next blank command line. I tried resizing the image size as well. Would you be able to help me with this?

Adrian Rosebrock

December 10, 2016 at 7:21 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-413110)

If your Python script exited without error then I would debug the script by doing:

```
print(circles)
```

My bet is that no circles were actually detected and you need to tune the parameters to HoughCircles.

Miral Desai

February 20, 2017 at 5:38 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-418087)

Hi….

I want to detect eyeball circle. I dumped the same script. But i got the error message as follows:

output=image.copy()
AttributeError: 'None Type' object has no attribute 'copy'

Adrian Rosebrock

February 20, 2017 at 7:37 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-418091)

It sounds like your image was not loaded from disk correctly, perhaps due to your command line arguments (i.e., path to an invalid file). Please take a look at **this blog post (https://www.pyimagesearch.com/2016/12/26/opencv-resolving-nonetype-errors/)** for more information on

resolving "NoneType" errors.

Satvik

March 14, 2017 at 2:12 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-420205)

I'm doing a project on smar fuel dispenser . Idea is to develop auto adjusting fuel dispenser . So can a get a code that will detect the fuel tank opening (circular) and which can give the position of it (X Y Z co ordinates) plz help me with this . Thank you

Uwe

March 15, 2017 at 4:57 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-420338)

Hello Adrian,
If I have a Pi camera image, how is a circular detection possible?
Thanks for the answer
Uwe Reinersmann

Adrian Rosebrock

You would need to **access the Raspberry Pi camera (https://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/)** and then apply circle detection to each frame.

quiqueapolo

March 20, 2017 at 1:09 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-420817)

Hi Adrian:

I want to count the number of oranges from this image: **http://cmapspublic.ihmc.us/rid=1KMVN1NVZ-1FDS32V-1PMG/naranjos-con-naranjas.jpg (http://cmapspublic.ihmc.us/rid=1KMVN1NVZ-1FDS32V-1PMG/naranjos-con-naranjas.jpg)**

Is it posiible? Could you give me a demostration if you want.

Adrian Rosebrock

March 21, 2017 at 7:17 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-420885)

I would use a combination of detecting the color orange with a **Histogram of Oriented Gradients**

Eric

March 23, 2017 at 7:25 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-421080)

HI Adrian, been enjoying you Practical Python and OpenCV book. It was a great starter. Now I'm trying to combine it with some of your blog examples. In the soda can example above, how would you create a mask for the outside of the circle? I could probably use some sort of fill to paint the outside but I would like to be able to create a mask so I can switch the mask on and off.

Keep up the awesome work!

Adrian Rosebrock

March 25, 2017 at 9:33 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-421183)

I would first detect the circle in the image. Then, create a mask for the circle. Then invert the mask. This will give you a mask for the non-circle region of the image.

May 9, 2017 at 2:56 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-424854)

is there any difference if i read image, apply blur, canny edge then houghcircle?
I am not getting importance of param1 for providing internal canny? When i used internal canny i was not able to detect circle matrix.
When i applied canny first then hough circle, false rejection was reduced. What does it make a sense?

Adrian Rosebrock

May 11, 2017 at 8:58 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-424979)

It really depends on your input images. If your images are fairly “clean” already you can skip the blurring step. Applying the Canny edge detector further helps clean up the image, giving you only the binary edges. If your edges are well defined, it will improve the accuracy of the circle detector.

Mono

June 7, 2017 at 5:48 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-426905)

I am facing some challenges to detect a tennis ball, o you have any tutorial or information related to parameter Tuning ?

Adrian Rosebrock

June 9, 2017 at 1:45 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-427009)

Hi Mono — instead of using Hough Circles to detect the tennis ball, why not detect the yellow of the tennis ball? That would be easier than working with Hough Circles.

Debajyoti Dutta

June 13, 2017 at 8:58 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-427232)

Sir,

Adrian Rosebrock

June 13, 2017 at 10:51 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-427239)

You would simply need to use the `cv2.circle` function. You can read about the function **here (http://docs.opencv.org/2.4/modules/core/doc/drawing_functions.html#circle)**. I also cover how to draw circles around contours inside **Practical Python and OpenCV (https://www.pyimagesearch.com/practical-python-opencv/)**.

Kevin Judd

June 16, 2017 at 2:11 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-427489)

Hey, thanks for the tutorial. I'm working on a project that requires the detection of a specific circle in an image that could potentially have multiple circles. I set the minimum distance between center points (minDist) to be equal to the length of the image so that the function would only be allowed to find a single circle despite the image having multiple circles in it. So I was wondering how the HoughCircles function decides which circle to output. Does it take the one that most closely resembles a circle, or maybe the circle with the greatest radius? The 8 circle example is a

Adrian Rosebrock

June 20, 2017 at 11:19 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-427777)

It chooses the outside circle instead of the inside circle due to the `minDist` parameter. It will also filter circles based on any supplied maximum/minimum radius.

Pranita

June 22, 2017 at 3:41 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-427852)

Hey Adrian,
Thank you for this tutorial.
I am trying to detect ellipse shaped structures from images. I know there is function for ellipse detection in skimage and I also read your tutorial where you detect different shapes in an image. But my problem is I have images wherein the ellipse are not exact ellipse (like deformed ellipse). These ellipses are cells from some patient data and every patient image will have different size and shape of ellipses. Can it be still detected by hough transform?

I tried using hough_ellipse but does not work well. I also tried k-means clustering but do not get convincing results.

Adrian Rosebrock

June 22, 2017 at 9:26 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-427867)

If the objects you are trying to detect are “deformed ellipses”, then I would suggest extracting contours and then computing the aspect ratio, extent, solidity, etc. These can be used to filter your shapes using a series of “if” statements. An example of using solidity/extent/aspect ratio to filter shapes can be found **here (https://www.pyimagesearch.com/2015/05/04/target-acquired-finding-targets-in-drone-and-quadcopter-video-streams-using-python-and-opencv/)**.

Rajnikant Sharma

July 8, 2017 at 1:35 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-429328)

Hi Adrian,
I am matlab developer, are we able to detect doors from following floorplan images using opencv?

https://www.ada.gov/archive/NPRM2008/images/plan2aaccessible.jpg

and

http://kesterhouse.com/interior/photos/floorplan_01.png
(http://kesterhouse.com/interior/photos/floorplan_01.png)

thanks

Adrian Rosebrock
July 11, 2017 at 6:47 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-429556)

Hi Rajnikant — I am not a MATLAB user. Are you using me how to solve this issue with MATLAB?

Evan
July 31, 2017 at 3:37 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-431246)

Hello, thanks for this awesome tutorial. Since this seems like a great way to do ball-tracking, I'm interested in your

Adrian Rosebrock

August 1, 2017 at 9:38 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-431303)

The reason is because the parameters of Hough Circles can be a real pain to tune in a case-by-case basis. Using simple color thresholding was a more accurate method to finding the ball and tracking it.

Mohamed O.

August 30, 2017 at 1:21 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-433658)

Hi Adrian,
Thanks for the great tutorials.
My question is: what kind of filters are helpful when you try to identify protruded circles made from the same material as the background, for example like the top surface of a child's building block?
I tried to convert the image first from color to grayscale, then applied a bilateral filter. I also tried to go one further step by converting the output of the bilateral filter to binary image, but it didn't help either.

Below is the code for the filters I used:

```
blurred = cv2.bilateralFilter(gray, 1, 60, 20)
thresh = cv2.threshold(blurred, 84, 255, cv2.THRESH_BINARY)[1]
```

The values I used in the code are based on tuning using different combinations of values, till I found out that these are the best match for my image.

Appreciate any recommendations to achieve the best possible result.

Thanks.

Adrian Rosebrock

August 31, 2017 at 8:34 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-433746)

Hi Mohamed — I would have to see an example image of these “protruded circles” to suggest a method.

Mohamed O.

September 2, 2017 at 10:28 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-433939)

I use this images as an example:
https://ibb.co/bEm3Wv (https://ibb.co/bEm3Wv)
and so far, I am only able to correctly identify the leftmost circle in the top row, & the 3 leftmost circles in the bottom row.
I am not sure why the code is not able to identify the other circles, but I assume it is due to low image quality. Is my assumption correct?
Thanks in advance for your advice.

Adrian Rosebrock
September 5, 2017 at 9:35 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-434142)

For low quality images with shadowing (which will be common with Lego images) using simple circle detection will likely be inaccurate. I would suggest using edge detection + contour processing, as detailed in **Practical Python and OpenCV (https://www.pyimagesearch.com/practical-python-opencv/)**. You might also want to consider training a simple circle detector using the framework discussed in **this blog post (https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/)**.

satinder

Hello sir, i would like to tell you that your tutorials are by far the best ones i have come across.
I would like to know how you can detect the color of a circular ring and also i wanted to know how i can know whether something has gone through the ring or not.THANK YOU

Adrian Rosebrock
September 18, 2017 at 1:59 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-435106)

It really depends on your particular setup. I would suggest using color thresholding to identify the actual ring to start.

siriusk
September 25, 2017 at 5:12 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-435617)

This algorithm is not bad when you've an image with only circle, but when you've an image with many edges, in my tests, it's not really efficient. There's a way to made this algorithm more robust?

I demonstrate how to recognize shapes based on their contour properties in **this post (https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/)**. I would suggest starting there if you have objects that have vertices.

Anonymous
October 8, 2017 at 11:03 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-436801)

Hello Adrian,
If I wanted to do this from a raspberry pi using live stream and not one image, how would the code change?

Adrian Rosebrock
October 9, 2017 at 12:24 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-436950)

I would suggest starting with **this post (https://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-**

Shoba

October 26, 2017 at 3:09 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-438718)

Hi
I want to detect the Tyre area in truck image and do color transformation over the area since the Tyre color close with shadow of image so it complicate the shadow removal task. Can you please guide how to detect the Tyre. How can i upload the image for your reference here.
Thanks and Regards,

Adrian Rosebrock

October 26, 2017 at 11:40 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-438756)

Hi Shoba — Feel free to post the images on **pasteboard (https://pasteboard.co/)** and provide a link here.

Shoba

Hi Adrian, Thanks for your reply but i cannot upload the image using the mentioned link suggest me how to do it.

Adrian Rosebrock

October 30, 2017 at 3:10 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-439205)

You can drag and drop images into the webpage. If that doesn't work, try **https://imgur.com/ (https://imgur.com/)**.

smtabatabaie

November 12, 2017 at 3:35 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-440307)

Hi Adrian, Thanks for the tutorial. I just had a question. Will this technique also work if circles in the image has perspective and are not completely in front view? So they are not complete circles and in perspective they will become ovals.
Thanks

Unfortunately no, this technique will rapidly fail once the viewing angle starts to change. The elliptical transform **inside scikit-image (v)** might be better in this particular case.

Rohit

December 7, 2017 at 8:23 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-442613)

hi Adrian,
Can you tell me how to detect a ring or a hoop from any viewing angle??

Thank in advance!!

greatdevaks (https://www.linkedin.com/in/greatdevaks)

November 21, 2017 at 6:54 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-441241)

Hey Adrian,

was thinking that what can be done to detect the concentric circles?

Waiting for your earliest response.

Thanks

Adrian Rosebrock

November 21, 2017 at 1:19 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-441264)

Yes, you can do this with concentric circles. Start with the largest possible radius. Detect the circle. Mask it out. Detect the next largest circle. Mask it out. Repeat until all circles have been detected.

Ahmad Afghan

December 15, 2017 at 5:12 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-443304)

i detected the circles but i aslo need to count these circles and print it on consol.

Adrian Rosebrock

December 15, 2017 at 8:18 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-443323)

All you need is:

```
print(len(circles))
```

amber

January 8, 2018 at 3:30 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-445875)

i m a beginner in python.. i am confused about argument parsing.. in your tutorial exactly where should i give the path to my image?in help or somewhere else?

Adrian Rosebrock

January 8, 2018 at 2:37 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-445943)

line arguments in the terminal. You do not need to update the code. I would suggest **reading up on command line arguments (https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/)** to help you get started.

Tanmay Fuse

January 26, 2018 at 5:11 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-447831)

How to crop hough circle??

Adrian Rosebrock

January 30, 2018 at 10:42 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-448255)

You can compute the bounding box of the rectangle and apply array slicing to extract it.

Marilia

July 7, 2019 at 1:19 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-

Could you help us with that? I tryed but is not working well. :-)

Adrian Rosebrock

July 10, 2019 at 9:54 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-525193)

I can't write the code for you but I can teach you. You should read **Practical Python and OpenCV (https://www.pyimagesearch.com/practical-python-opencv/)** to learn how to compute bounding box coordinates and extract ROIs from an image.

Ashu

February 5, 2018 at 6:16 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-449066)

The code seems to run only for squarish images..For example, this image does not work:
https://upload.wikimedia.org/wikipedia/commons/thumb/9/9d/DFAexample.svg/250px-DFAexample.svg.png (https://upload.wikimedia.org/wikipedia/commons/thumb/9/9d/DFAexample.svg/250px-DFAexample.svg.png)

And can you explain how to adjust the parameters for each image and on what basis? Is there a way to automate

Adrian Rosebrock

February 6, 2018 at 10:17 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-449251)

You might need to tune the parameters to the Hough circle detector in order for it to detect all circles in your images. Unfortunately there isn't a way to automate this. A more robust method may be to **train your own circle detector (https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/)**.

raj

February 25, 2018 at 1:06 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-451241)

hello, the above code is for tracking circular objects like what if we want to track object with different shape?

Adrian Rosebrock

February 26, 2018 at 1:56 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-451385)

There are a few ways to accomplish this, but mostly dependent on the dataset/project. What types of objects are

Hrishi

March 1, 2018 at 1:08 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-451643)

Hi!....your tutorials are very nice and informative.For my project i want to detect only red circles using raspberry and pi cam.Can you help me?

Adrian Rosebrock

March 2, 2018 at 10:48 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-451777)

Take a look at **this post (https://www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/)** on shape and color detection.

Juan Giraldo

March 15, 2018 at 12:09 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-453183)

First I would like to highlight how informative and creative are your tutorials on artificial vision, really are very entertenidos and didactic, I congratulate you !.

My question is can you count in a video the number of axles of a truck, 2 axles, 4 axles, 9?. I tried it but several inconvenients arise when trying to separate only the contours of the rim, detect several contours or not detect them with the function cv2.HoughCircles.

Adrian Rosebrock

March 19, 2018 at 5:48 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-453611)

Thank you for the kind words, Juan. I'm glad you are enjoying the blog!

As for your question, how are you attempting to detect each set of axels?

Juan Giraldo

March 20, 2018 at 1:06 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-453737)

Describing quickly what I have tried, it goes like this:

– I play a video of a truck parade and to this I subtract the fund.
– Then I perform two commands for Morphological Transformations (opening and closing) and make the video transmission clear.

– Then I look for contours with the command cv2.RETR_EXTERNAL

What happens is that many contours medetectan and only need those that are round to be able to count the axes of the truck (as the view is lateral each tire detected would represent an axis).

– Try to define the minimum area that should have the outline but it is not very accurate and I keep appearing many

– Then I would like to do it like in a tutorial of yours, I could observe how the trajectory of a green ball was followed and this is adapted for the black colored tires; The problem is that the tire color does not always have the same intensity and the circle on a trowel is not uniform, it's like a kind of washer.

With the function cv2.Canny (gray, 200, 300), I see well-defined circles in the transmission of the vine but I can

This I want to do is a hobby way to learn.

Thank you very much.

Adrian Rosebrock

March 22, 2018 at 10:15 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-453956)

Thank you for sharing the details Juan. I get the idea of what you are doing but it would be helpful to see some visuals. Would you be able to share any example images?

Mike Oliver

March 15, 2018 at 6:51 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-453204)

Good Morning!

What do I do?

Adrian Rosebrock

March 19, 2018 at 5:43 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-453606)

Hey Mike — it sounds like you'll need to detect the parameters to `cv2.HoughCircles`. The function can be a bit of a pain to use. Depending on your input images you may need to utilize machine learning to train your own circle detector.

Sai

April 5, 2018 at 4:14 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-455237)

Thanks a lot Adrian. Very well explained.

Adrian Rosebrock

Thanks Sai, I'm glad you enjoyed it

Mesco

June 6, 2018 at 10:31 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-467215)

Hi, thanks for tutorial.

OpenCV documentation says: "method – Detection method to use. Currently, the only implemented method is CV_HOUGH_GRADIENT , which is basically 21HT , described in [Yuen90]."

https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=houghcircles#houghcircles (https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=houghcircles#houghcircles)

So now I'm totally confused: why it is called "gradient"?

Also, despite of reading Yuen article I do not understand what is "accumulator"

inferno

June 16, 2018 at 5:51 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-468248)

how to resolve the problem of minimum distance parameter

Adrian Rosebrock

June 19, 2018 at 8:59 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-468558)

Unfortunately for the Hough Circles function it's a trial and error tuning process.

Malouke

July 12, 2018 at 5:45 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-470831)

hi thank you so much about topics .

i have a remark about topics the HoughCircles algorithmes it s iterative one .

less accurate.

my quesions can i ask you to do the same things but with multipls balls more than 10 and same color of balls you will see its hard to detect the circle shape.

Thank s

Adrian Rosebrock

July 13, 2018 at 5:07 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-470919)

You are correct, once you start creating more complex backgrounds Hough Circles will fail. For more complex objects you may want to consider training a custom object detector. For well defined shapes such as circles **HOG + Linear SVM (https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/)** would be a good start.

Abi

August 12, 2018 at 5:29 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-474060)

This tutorial is a great help. However, as a beginner in python programming, may I ask for help on how I can create a

Adrian Rosebrock

August 15, 2018 at 8:59 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-474408)

I'm not sure what exactly you're trying to graph, but matplotlib tends to be a good library for plots and graphs.

Abi

August 24, 2018 at 9:14 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-475413)

May I ask for help on how I can store all the coordinates of the centers of the circles detected?

Adrian Rosebrock

August 30, 2018 at 9:43 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-476003)

Store as in write them to disk? You could use a CSV file, text file, JSON file, or whatever file format you wish.

hs

September 2, 2018 at 11:06 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-476304)

Hi, can i ask for help how do i make the camera capture once circle detected? thx

---

Adrian Rosebrock

September 5, 2018 at 9:05 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-476723)

What does “capture” mean? Save a single frame to disk? Save a video clip?

Manigandan K P

September 4, 2018 at 1:44 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-476481)

Hello Sir, how to convert python file to Exe Application

Zamra

September 25, 2018 at 3:28 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-479571)

Hi, Is there any ways to find the intersection points of two detected circles using the x,y and radius of those 2 circles. I am trying to extract features from a 2 set Venn diagram therefore there intersection points are needed.

Zamra

September 25, 2018 at 3:33 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-479573)

Hi, Are there ways to store the detected circles to separate variables?

Adrian Rosebrock

October 8, 2018 at 12:46 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-481318)

Hey Zamra, I'm not sure what you mean by "separate variables" — could you elaborate?

Andrea

October 3, 2018 at 12:34 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-480634)

Could you post a simple example of how to use the MinRadius and MaxRadious parameters please?

Stefano Ceresa

October 6, 2018 at 12:35 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-480935)

Hello,
I coded everything and there are no errors but when I send the program from the shell (I use a mac) I do not see anyout

---

Adrian Rosebrock

October 8, 2018 at 9:41 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-481168)

Are you executing via the command line or via a Python shell itself? Try executing the script via the command line.

Aashna

October 31, 2018 at 5:00 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-484821)

Hi Adrian,
What if we want to detect just the region between two concentric circles and save that as another image
Thanks

Adrian Rosebrock

November 2, 2018 at 7:22 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-485014)

You can use the “cv2.imwrite” function to save an image/ROI to disk.

Srinivasab

November 21, 2018 at 1:45 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-488328)

I want to measure the multicircles.its just like an example .. olympic rings are not interlinked.is it possible.

Thanks in advance
Srinivasan

Adrian Rosebrock
November 25, 2018 at 9:40 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-489043)

What specifically are you trying to measure? Just the size in pixels? Or the size in real-world measurement units?

Jimmy
November 28, 2018 at 12:30 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-489629)

Hi, could you share a c++code with this same functionnality please ?

Best regards

Sorry, I only provide Python code here on the PyImageSearch blog.

sathithya yogi
February 11, 2019 at 10:53 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-500826)

how to contour two circle with same center, i want to measure the diameter for the ring

Akshaya Balamurugan
February 28, 2019 at 11:31 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-504075)

Hi Adrian,

Thank You for this awesome tutorial once again. I am trying to identify one circle from an image in which the circle is

Could you please let me know what would be the case here and what I should be looking at to get this corrected.

Looking forward to your response.

Thank You !

Adrian Rosebrock

March 1, 2019 at 5:26 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-504109)

Do you have any example images of what your input and output look like? It's a bit hard to give any suggestions to this problem without seeing them first.

Hossary

April 28, 2019 at 3:56 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-515089)

Hi Adrian,

quarter circle) with this code , any suggestion to use Hough circle to detect them ?
Thank You for your help !

Valerio
August 2, 2019 at 7:55 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-529881)

Hi Adrian,
I was wondering would it be easier to use contours and look for approxPolyDP with many vertices compared to HoughCircles where you need to tune it quite well unless you want a number of false positives.
Best regards

Adrian Rosebrock
August 7, 2019 at 12:34 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-532720)

In controlled environments you could certainly do that. The problem becomes if you end up detecting blobs that are essentially noise. In that case it would be falsely labeled as a circle.

Antonios

August 15, 2019 at 6:02 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-538346)

Hey Adrian.
What if in a picture there are both circles and ellipses, and we need only the total circles.
How to separate them and get only circles ?
Continue the great job.

Adrian Rosebrock

August 16, 2019 at 5:31 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-539187)

Take a look at the **scikit-image library (https://scikit-image.org/docs/dev/auto_examples/edges/plot_circular_elliptical_hough_transform.html)**. They have methods that can detect both circles and ellipses.

Sher

September 27, 2019 at 9:30 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-

Thank you so much for such a wonderful tutorial. I am trying to use this code for y project. The image is created from lidar point cloud. It represents diameters of trees, bushes and some other noise. I would like to use this code for detecting diameter of trees as circles. Would this code be helpful? It is running with your sample images but, for my images it run well and created false circles first. Then, now it is running, but no any output? I tried to change minDist with no success. Sorry for multiple questions, do you have any tutorial for object detection in Lidar point cloud? Thanks a lot.

Codie

November 15, 2019 at 3:07 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-574021)

Hey Adrian,

I was wondering if there is a way to count how many hough circles are found and then print that number on the screen.

Thanks

Yes, just do:

```
print(len(circles))
```

tuhuynh

December 25, 2019 at 9:41 pm (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-616729)

Dear Adrian,

i wonder if HoughCircle can detect ellipse? i want to detect ellipse in picture and find the diameter but i dont know how to solve this problem. Do you have any idea about this problem?

Adrian Rosebrock

December 26, 2019 at 9:49 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-617862)

I would recommend you use **scikit-image's ellipse detection (https://scikit-**

sahar tabrizi

January 2, 2020 at 12:10 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-622255)

Hi Adrian,

Thanks very much for a very helpful tutorial.

I have a question regarding sub-pixel accuracy! why the (x, y) coordinates and radius of the circles need to be converted to integers?

Thanks!
Sahar

Safa Pary

March 4, 2020 at 11:52 am (https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/#comment-765444)

well?

Before you leave a comment...

Hey, Adrian here, author of the PyImageSearch blog. I'd love to hear from you; however, I have made the decision to no longer offer free 1:1 help over blog post comments. I simply do not have the time to moderate and respond to them all.

To that end, myself and my team are doubling down our efforts on supporting our paying customers, writing new books and courses, and authoring high quality Computer Vision, Deep Learning, and OpenCV content for you to learn from.

I'd be happy to help you with your question or project, but **I have to politely ask you to purchase one of my books or courses first. (https://www.pyimagesearch.com/books-and-courses/)**

Why bother becoming a PyImageSearch customer?

- You'll receive a **great education** through my premium content.

- You'll receive a **guaranteed response** from myself and my team.
- You'll be able to **confidently and successfully** apply Computer Vision, Deep Learning, and OpenCV to your projects.

Click here to see my full catalog of books and courses. (https://www.pyimagesearch.com/books-and-courses/) Take a look and I hope to see you on the other side!

Similar articles

OBJECT DETECTION TUTORIALS

November 9, 2015

(https://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/)

IMAGE SEARCH ENGINE BASICS

Building an Image Search Engine: Searching and Ranking (Step 4 of 4)

February 24, 2014

(https://www.pyimagesearch.com/2014/02/24/building-image-search-engine-searching-ranking-step-4-4/)

OPENCV TUTORIALS TUTORIALS

How to install OpenCV 4 on Ubuntu

August 15, 2018

(https://www.pyimagesearch.com/2018/08/15/how-to-install-opencv-4-on-ubuntu/)

You can learn Computer Vision, Deep Learning, and OpenCV.

Get your FREE 17 page Computer Vision, OpenCV, and Deep Learning Resource Guide PDF. Inside you'll find my hand-picked tutorials, books, courses, and libraries to help you master CV and DL.

Topics

Machine Learning and Computer Vision
(https://www.pyimagesearch.com/category/machine-learning-2/)

Dlib Library (https://www.pyimagesearch.com/category/dlib/)

Embedded/IoT and Computer Vision (https://www.pyimagesearch.com/category/embedded/)

Face Applications (https://www.pyimagesearch.com/category/faces/)

Image Processing (https://www.pyimagesearch.com/category/image-processing/)

Interviews (https://www.pyimagesearch.com/category/interviews/)

Keras (https://www.pyimagesearch.com/category/keras/)

Optical Character Recognition (OCR) (https://www.pyimagesearch.com/category/optical-character-recognition-ocr/)

Object Detection (https://www.pyimagesearch.com/category/object-detection/)

Object Tracking (https://www.pyimagesearch.com/category/object-tracking/)

OpenCV Tutorials (https://www.pyimagesearch.com/category/opencv/)

Raspberry Pi (https://www.pyimagesearch.com/category/raspberry-pi/)

Books & Courses

FREE CV, DL, and OpenCV Crash Course (https://www.pyimagesearch.com/free-opencv-computer-vision-deep-learning-crash-course/)

Practical Python and OpenCV (https://www.pyimagesearch.com/practical-python-opencv/)

Deep Learning for Computer Vision with Python (https://www.pyimagesearch.com/deep-learning-computer-

PyImageSearch

Get Started (https://www.pyimagesearch.com/start-here/)

OpenCV Install Guides (https://www.pyimagesearch.com/opencv-tutorials-resources-guides/)

About (https://www.pyimagesearch.com/about/)

FAQ (https://www.pyimagesearch.com/faqs/)

Blog (https://www.pyimagesearch.com/topics/)

PyImageSearch Gurus Course (https://www.pyimagesearch.com/pyimagesearch-gurus/)

Raspberry Pi for Computer Vision (https://www.pyimagesearch.com/raspberry-pi-for-computer-vision/)

Privacy Policy (https://www.pyimagesearch.com/privacy-policy/)

(https://www.facebook.com/pyimagesearch) **(https://twitter.com/PyImageSearch)** **(http://www.linkedin.com/pub/adrian-rosebrock/2a/873/59b)** **(https://www.youtube.com/channel/UCoQK7OVcIVy-nV4m-SMCk_Q/videos)**