

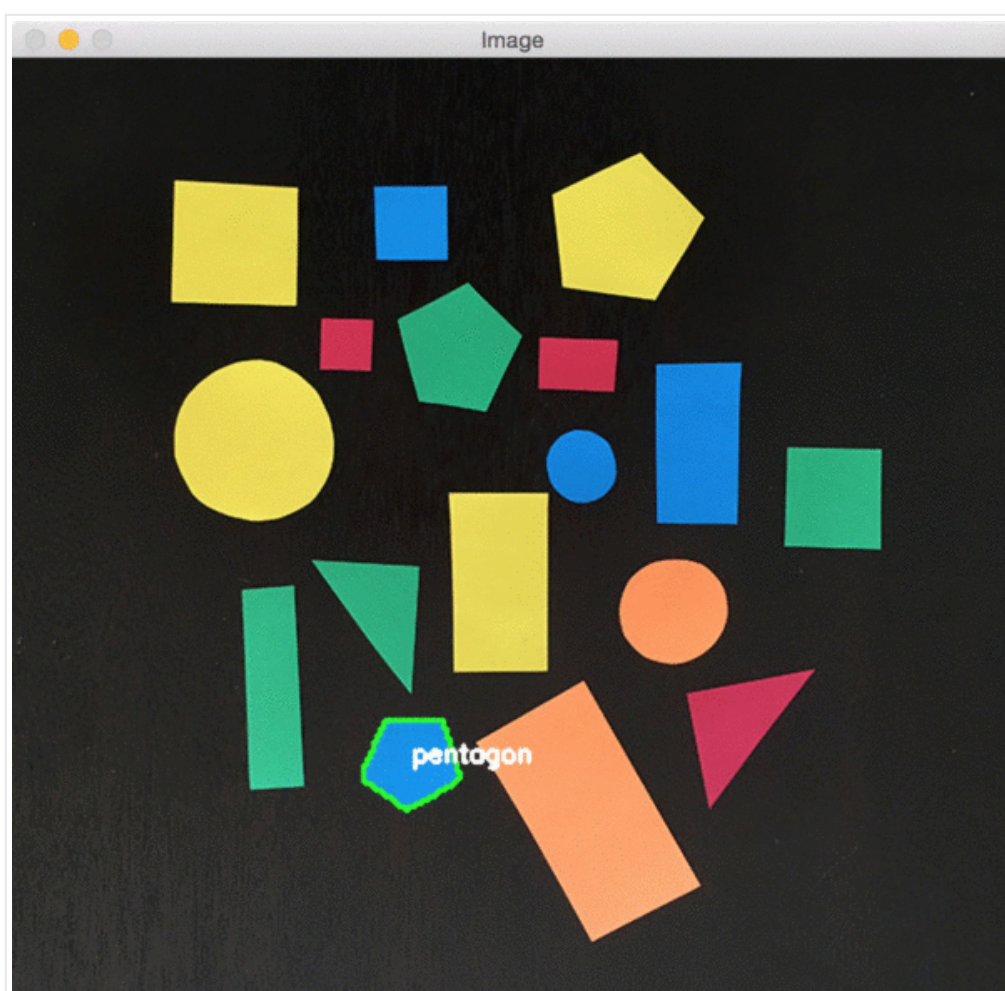


OpenCV shape detection

by **Adrian Rosebrock** on February 8, 2016 in **Image Processing, OpenCV 3, Tutorials**



[Click here to download the source code to this post.](#)



This tutorial is the second post in our three part series on *shape detection and analysis*.

Last week we learned how to compute the [center of a contour](#) using OpenCV.

Today, we are going to leverage contour properties to actually *label* and *identify* shapes in an image, just like in the figure at the top of this post.

**Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning**

Looking for the source code to this post?
Jump right to the downloads section.

OpenCV shape detection

Before we get started with this tutorial, let's quickly review our project structure:

```
OpenCV shape detection Shell
1 |--- pyimagesearch
2 |   |--- __init__.py
3 |   |--- shapedetector.py
4 |--- detect_shapes.py
5 |--- shapes_and_colors.png
```

As you can see, we have defined a `pyimagesearch` directory which will store our implementation of the `shapedetector.py` which will store our implementation of the

Finally, we have the `detect_shapes.py` driver script which will load the image, detect shapes, and then perform shape detection and identification.

Before we get started, make sure you have the `imutils` library installed. `imutils` is a small OpenCV convenience functions that we'll be using later in the tutorial.

```
OpenCV shape detection
1 $ pip install imutils
```

Defining our shape detector

The first step in building our shape detector is to write the logic for the `ShapeDetector` class.

Let's go ahead and define our `ShapeDetector` class. Open the `shapedetector.py` file and paste the following code:

```
1 # import the necessary packages
2 import cv2
3
4 class ShapeDetector:
5     def __init__(self):
6         pass
7
8     def detect(self, c):
9         # initialize the shape name and approximate the contour
10        shape = "unidentified"
11        peri = cv2.arcLength(c, True)
12        approx = cv2.approxPolyDP(c, 0.04 * peri, True)
```

Line 4 starts the definition of our `ShapeDetector` class. We'll skip the `__init__` constructor here since nothing needs to be initialized.

We then have our `detect` method on **Line 8** which will take a contour (the `c` argument) and return the outline of the shape we are trying to identify.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

In order to perform shape detection, we'll be using *contour approximation*.

As the name suggests, contour approximation is an algorithm for reducing the number of points in a curve with a reduced set of points — thus the term *approximation*.

This algorithm is commonly known as the [Ramer-Douglas-Peucker](#) algorithm, or simply the split-and-merge algorithm.

Contour approximation is predicated on the assumption that a curve can be approximated by a *series of short line segments*. This leads to a resulting approximated curve that consists of a subset of points that were defined by the original curve.

Contour approximation is actually already implemented in OpenCV.

In order to perform contour approximation, we first obtain the actual contour points and then approximate by constructing the actual contour approximation (Liu, 2015).

Common values for the second parameter to `cv2.approxPolyDP` are 1% and 5% of the original contour perimeter.

Note: Interested in a more in-depth look at contour approximation? Check out my [PyImageSearch Gurus course](#) where I discuss computer vision, contours and connected-component analysis in detail.

Given our approximated contour, we can move on to shape detection.

OpenCV shape detection

```

14     # if the shape is a triangle, it will have 3 vertices
15     if len(approx) == 3:
16         shape = "triangle"
17
18     # if the shape has 4 vertices, it could be a square or
19     # a rectangle
20     elif len(approx) == 4:
21         # compute the bounding box of the shape
22         # bounding box to compute the aspect ratio
23         (x, y, w, h) = cv2.boundingRect(approx)
24         ar = w / float(h)
25
26         # a square will have an aspect ratio that is approximately
27         # equal to one, otherwise, the shape is a rectangle
28         shape = "square" if ar >= 0.95 and ar <= 1.05 else "rectangle"
29
30     # if the shape is a pentagon, it will have 5 vertices
31     elif len(approx) == 5:
32         shape = "pentagon"
33
34     # otherwise, we assume the shape is a circle
35     else:
36         shape = "circle"
37
38     # return the name of the shape
39     return shape

```

It's important to understand that a contour consists of a list of points. To determine the shape of an object, we iterate over the entries in this list to determine the shape of an object.

For example, if the approximated contour has *three* vertices, we can determine the shape is a triangle.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning

If a contour has *four vertices*, then it must be either a *square* or a *rectangle* (**Line 20**). To determine which, we compute the aspect ratio of the shape, which is simply the width of the contour bounding box divided by the height (**Lines 23 and 24**). If the aspect ratio is ~ 1.0 , then we are examining a square (since all sides have approximately equal length). Otherwise, the shape is a rectangle.

If a contour has *five vertices*, we can label it as a *pentagon* (**Line 31 and 32**).

Otherwise, by process of elimination (in context of this example, of course), we can make the assumption that the shape we are examining is a *circle* (**Lines 35 and 36**).

Finally, we return the identified shape to the calling method.

Shape detection with OpenCV

Now that our `ShapeDetector` class has been defined

OpenCV shape detection

```
1 # import the necessary packages
2 from pyimagesearch.shapedetector import ShapeDetector
3 import argparse
4 import imutils
5 import cv2
6
7 # construct the argument parse and parse the arguments
8 ap = argparse.ArgumentParser()
9 ap.add_argument("-i", "--image", required=True,
10               help="path to the input image")
11 args = vars(ap.parse_args())
```

We start off on **Lines 2-5** by importing our required packages and the implementation of the `ShapeDetector` class from the

Lines 8-11 handle parsing our command line arguments, where `args` is the path to where the image we want to process is located.

Next up, let's pre-process our image:

OpenCV shape detection

```
13 # load the image and resize it to a smaller factor so that
14 # the shapes can be approximated better
15 image = cv2.imread(args["image"])
16 resized = imutils.resize(image, width=300)
17 ratio = image.shape[0] / float(resized.shape[0])
18
19 # convert the resized image to grayscale, blur it slightly,
20 # and threshold it
21 gray = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)
22 blurred = cv2.GaussianBlur(gray, (5, 5), 0)
23 thresh = cv2.threshold(blurred, 60, 255, cv2.THRESH_BINARY)[1]
24
25 # find contours in the thresholded image and initialize the
26 # shape detector
27 cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
28                       cv2.CHAIN_APPROX_SIMPLE)
29 cnts = imutils.grab_contours(cnts)
30 sd = ShapeDetector()
```

First, we load our image from disk on **Line 15** and resize it to a smaller factor so that the shapes can be approximated better. We then compute the ratio of the old height to the new resized height on **Line 17**.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning

tutorial.

From there, **Lines 21-23** handle converting the resized image to grayscale, smoothing it to reduce high frequency noise, and finally thresholding it to reveal the shapes in the image.

After thresholding, our image should look like this:

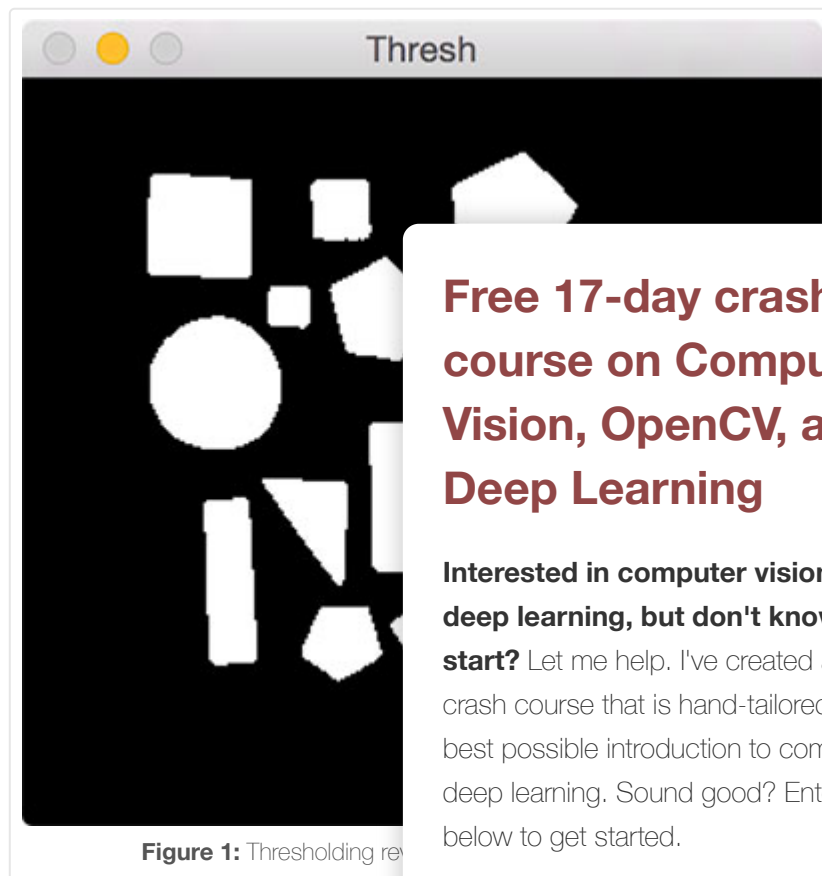


Figure 1: Thresholding result

Notice how our image has been *binarized* — the shapes are white on a black background.

Lastly, we find contours in our binary image, handle each contour, and use `cv2.findContours` based on our OpenCV version, and finally initialize our `ShapeDetector` (**Lines 27-30**).

The last step is to identify each of the contours:

OpenCV shape detection

Python

```

32 # loop over the contours
33 for c in cnts:
34     # compute the center of the contour, then detect the name of the
35     # shape using only the contour
36     M = cv2.moments(c)
37     cX = int((M["m10"] / M["m00"]) * ratio)
38     cY = int((M["m01"] / M["m00"]) * ratio)
39     shape = sd.detect(c)
40
41     # multiply the contour (x, y)-coordinates by the resize ratio,
42     # then draw the contours and the name of the shape on the image
43     c = c.astype("float")
44     c *= ratio
45     c = c.astype("int")
46     cv2.drawContours(image, [c], -1, (0, 255, 0), 2)
47     cv2.putText(image, shape, (cX, cY), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))

```

Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning


```

48         0.5, (255, 255, 255), 2)
49
50     # show the output image
51     cv2.imshow("Image", image)
52     cv2.waitKey(0)

```

On **Line 33** we start looping over each of the individual contours. For each of them, we compute the center of the contour, followed by performing shape detection and labeling.

Since we are processing the contours extracted from the *resized image* (rather than the original image), we need to multiply the contours and center (x, y)-coordinates by our `ratio` (**Lines 43-45**). This will give us the correct (x, y)-coordinates for both the contours and centroid of the original image.

Lastly, we draw the contours and the labeled shape on our image (**Lines 44-48**), followed by displaying our results (**Lines 51 and 52**).

To see our shape detector in action, just execute the

OpenCV shape detection

```
1 $ python detect_shapes.py --image shapes_a
```

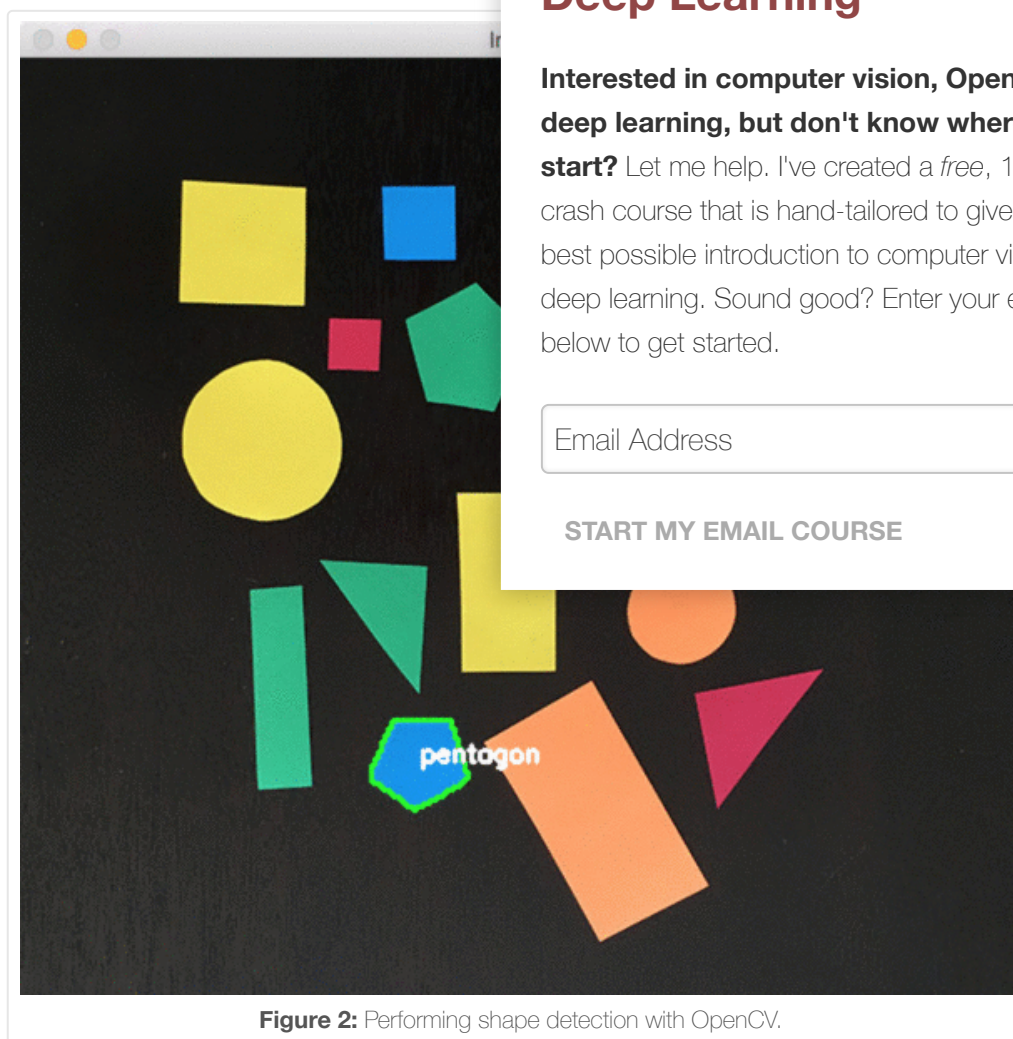


Figure 2: Performing shape detection with OpenCV.

As you can see from the animation above, our script loops over each of the shapes individually, performs shape detection on each one, and then draws the name of the shape on the object.

Summary

**Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning**

In today's post blog, we learned how to perform shape detection with OpenCV and Python.

To accomplish this, we leveraged *contour approximation*, the process of reducing the number of points on a curve to a more simple *approximated* version.

Then, based on this contour approximation, we examined the number of vertices each shape has. Given the vertex count, we were able to accurately label each of the shapes.

This lesson is part of a three part series on shape detection and analysis. Last week we covered how to compute the *center* of a contour. Today we covered shape detection with OpenCV. And next week we'll discuss how to **label the actual color of a shape** using color channel statistics.

Be sure to enter your email address in the form below **won't want to miss it!**

Downloads:

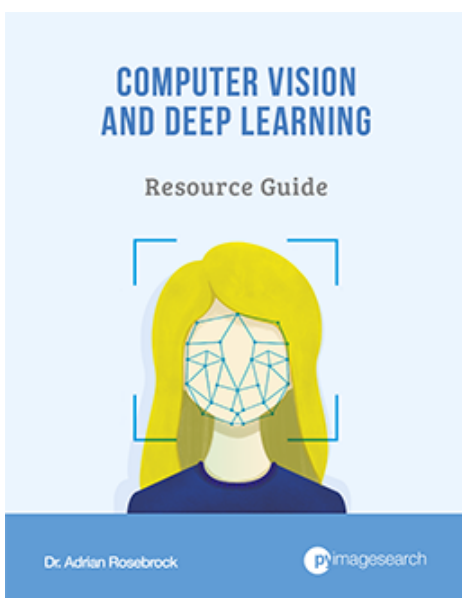


If you would like to download the email address in the form below you a **FREE 17-page Resource Learning**. Inside you'll find my help you master CV and DL! So you the code immediately!

Email address:

DOWNLOAD THE CODE!

Resource Guide (it's totally free).



Enter your email address below to get my **free 17-page Computer Vision, OpenCV, and Deep Learning Resource Guide PDF**. Inside you'll find my hand-picked tutorials, books, courses, and Python libraries to help you master computer vision and deep learning!

DOWNLOAD THE GUIDE!

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning

approximate contours, contour properties, contours, find contours, shapes[< OpenCV center of contour](#)[Determining object color with OpenCV >](#)

198 Responses to *OpenCV shape detection*

**leena** February 9, 2016 at 5:59 am #

REPLY

Why it is scanning and labeling from bottom

How to to scan and label top to bottom?

**Adrian Rosebrock** February 9, 2016 at 1:00 pm #That is how the `cv2.findContours` works. You can also use `cv2.findContours` contours, [see this post](#).**leena** February 17, 2016 at 4:57 am #

Thanks Adrian. It worked and I am able to

Please help me in identifying lines connected to each other. I want to know how to identify lines connected with line/arrow

with regards.

**leena** February 9, 2016 at 6:11 am #

REPLY

I have done the same with

$$\text{shape factor} = \text{area} / (\text{peri} * \text{peri})$$
$$\text{if shapefactor} \geq 0.06 \text{ and shapefactor} = 0.0484 \text{ and shapefactor} = 0.050 \text{ and shapefactor} = 0.032 \text{ and shapefactor} = 0.95 \text{ and ar} \leq 1.05 \text{ else "rectangle"}$$
**Adrian Rosebrock** February 9, 2016 at 3:55 pm #

REPLY

Is there a particular reason you are taking the ratio of the area to the perimeter squared? It seems to make the rule more complicated.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

**leena** February 10, 2016 at 11:11 pm #

REPLY ↩

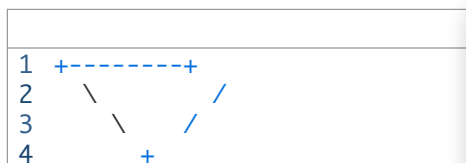
Actually I do not know the reason, just it got solved my problem, so I took it. You or somebody can help me understanding this and the better solution .

Thanks

**bitflip** June 21, 2016 at 11:30 am #

REPLY ↩

Given you have following triangle:



The bounding rect of it would have aspect ratio of 1.0. So, better take the area() of the contour when the error is too high → rectangle.

**Vincent** February 18, 2016 at 3:41 pm #

Hi Adrian,

First and foremost, thank you for this excellent tutorial!

I have used the logic here to detect red triangles in a class to identify only triangles. I am able to successfully detect them. <http://imgur.com/6Z9CnBA>

I've noticed that sometimes a very messy contour will be detected as a triangle. <http://imgur.com/4a06psM>

What would be a good way to tweak this?

<http://pastie.org/10727912>

<http://pastie.org/10727915>

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

**Adrian Rosebrock** February 18, 2016 at 4:05 pm #

REPLY ↩

Keep in mind that the code is only as good as the images that you put into it. The code detailed in this post assumes simple shapes that can be recognized utilizing contour properties. For more advanced shapes, or shapes that have substantial variances in how they appear (such as noisy contours), you might need to [train your own custom object detector](#).

Anyway, the reason sometimes even messy contours get classified differently is due to the contour approximation. Play around with the percentage of points to keep when approximating the differences.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

**Alfarabi** December 10, 2018 at 4:00 am #

REPLY ↩

Hi vincent, your shapedetector class to identify triangles is very usefull for me. Can you show me the source of code?

**Peng** March 4, 2016 at 7:49 pm #

REPLY ↩

Hi Adrian,

Thank you for making this. A little feedback on the image file.

I notice that if using a .jpg file as the source, the mor

It report an error :

```
cntX = int(M["m10"] / M["m00"])
```

```
ZeroDivisionError: float division by zero
```

Any ideas on this?

Thanks

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

**Adrian Rosebrock** March 6, 2016 at 9:2

Version version of OpenCV and Python

In either case, you can resolve the issue by doing

```
1 if M["m00"] > 0:
2     # ... continue to process the co
```

This **if** statement will take care of the divide by z

Alternatively, you can add a tiny value to the moment to ensure there is no divide by zero problem:

```
cX = (M["c10"] / (M["m00"] + 1e-7))
```

**saurabh** November 18, 2017 at 4:27 am #

REPLY ↩

can you please elaborate this .. i didn't get it..

```
=====
cX = int((M["m10"] / M["m00"] + 1e-7) * ratio)
ZeroDivisionError: float division by zero
=====
showing this error
```

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

**Adrian Rosebrock** November 18, 2017 at 8:07 am #

REPLY ↩

My original comment is missing a parenthesis:

$$M["c10"] / (M["m00"] + 1e-7)$$

Notice how the addition is done before the divide.

**Euan** March 10, 2016 at 10:00 pm #

REPLY ↩

Hi Adrian,

Firstly thanks for a great tutorial and site. I'm a mecha and have managed to get openCV 3.0 and python2.7 installed. <https://www.pyimagesearch.com/2015/06/22/install-opencv-3-0-on-ubuntu-14-04-lts/>

In order to get this code running on my setup, I need to solve some cast problems on line 43 "c *=ratio". I believe this is possible how it worked when you wrote this tutorial. Is this the case?

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

**Adrian Rosebrock** March 13, 2016 at 10:21 am #

Interesting, Brandon mentioned this issue. Are you using NumPy?

**brandon** March 11, 2016 at 4:20 pm #

Adrian, great stuff. I've learned a lot from you. I'm working through this post for now, and I'm getting the error message below.

Python

```
1 File "detect_shapes.py", line 46, in
2   c *= ratio
3 TypeError: Cannot cast ufunc multiply output from dtype('float64') to dtype('int32')
```

**Adrian Rosebrock** March 13, 2016 at 10:21 am #

REPLY ↩

I personally haven't seen this error message before. Can you let me know which OpenCV and NumPy versions you are using?

**Brandon** March 14, 2016 at 12:28 pm #

numpy is 1.10.4, and it happens with all versions of opencv (I've tried all thanks to another of your tutorials) under Pyth

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

The workaround was simply to adjust data types pre and post multiplication:

```
c=c.astype(np.float_)  
c *= ratio  
c=c.astype(np.int32)
```

It works now.



Adrian Rosebrock March 14, 2016 at 3:15 pm #

REPLY ↩

Thanks for the tip Brandon! I'll be sure to dive into this more. I'm using NumPy 1.9.3, so perhaps this is an issue with NumPy



Ahmed Abdeldaim March 24, 2016 at 10:30

Great work Mr. Adrian
but is there a way to make the selection more softer,
or this is the best result??



Adrian Rosebrock March 24, 2016 at 5

Absolutely, you just need to apply *contour*
few blog posts, [but I would start with this one](#).



Ahmed Abdeldaim March 26, 201

Thanks for your help.



darshan March 26, 2016 at 2:09 am #

REPLY ↩

how to install imutils module
I used pip install imutils I'm getting error



Adrian Rosebrock March 27, 2016 at 9:14 am #

REPLY ↩

Please see the "OpenCV shape detection" section of this blog post. You just need to use **pip**

```
$ pip install imutils
```

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



firoz khan April 23, 2016 at 10:42 am #

REPLY ↩

hi adrian it is only detcting one pentagon and nothing else



Adrian Rosebrock April 25, 2016 at 2:09 pm #

REPLY ↩

Make sure you click on the window and press any key on your keyboard — this will advance the script. Right now a keypress is required after each detection.



Diego Fernando Barrios April 29, 2016 at 1:00 pm #

Good afternoon!

Thanks very much for this tutorial, you're doing a great job!

Friend, I have a problem with contour detection, when I have a black background, I take the image from USB camera.

The python scripts should recognize (9 "nine" rectangles).

Sorry for the writing, my english is not so good.

I would like that you can help me. I'm working in my work.

Thanks very much!

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Adrian Rosebrock April 30, 2016 at 3:50 pm #

Depending on your image, this could be a problem. You might want to make sure that after thresholding your 9 rectangles have a good contour approximation and see how many points are returned by the approximation. You might need to tweak the `cv2.approxPolyDP` parameters.



itai May 8, 2016 at 4:24 am #

REPLY ↩

Hey Adrian,

I was wondering why did you use the Ramer-Douglas-Peucker algorithm to reduce the set of points, instead of using the convex hull ?

Thanks



Adrian Rosebrock May 8, 2016 at 8:12 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

The contour approximation algorithm and Convex Hull algorithm are used for two separate purposes. As the name implies, contour approximation is used to reduce the number of points along a contour by "simplifying" the contour based on a percentage of the perimeter. Your resulting contour approximation is this a simplification of the shape by utilizing points that are *already* part of the shape.

The convex hull on the other hand is the smallest convex set that contains all points along the contour — it is, by definition, not a simplification of the contour shape and the resulting convex hull actually contains points that are not part of the original shape.

In this case, I used contour approximation because I wanted to reduce the number of (x, y)-coordinates that comprise the contour, while ensuring that all points in the *resulting approximation* were also part of the *original shape*.



Armin June 30, 2016 at 2:56 pm #

Hello Adrian

thanks for tutorial

I want to show detected shapes in seperate windows
also tried cropping them (using ROI) but I didn't able to
tnx



Adrian Rosebrock July 1, 2016 at 2:59 pm #

Hey Armin — you're on the right track. You can use `cv2.imshow` function on each frame to display the image. I have a post as well as [Practical Python and OpenCV](#).



Alex Hopper July 24, 2016 at 5:04 pm #

Hello,

I'm new openCV-Python user.. I have a question about it:

I have a database containing pre-processed images by kinect, and I need use Deep Learning to analyse these images. Is there a best way to start?

<https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/>

Can I use it to start?

Thanks.



Adrian Rosebrock July 27, 2016 at 2:36 pm #

What type of images are you working with? Kinect. Are they depth images? RGB images?

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

REPLY ↩



Neal July 27, 2016 at 8:58 am #

REPLY ↩

hi

I'm looking for advice in shape detection. I want to use a camera to detect different kinds of shapes on a microcontroller. What would be the best method to approach this?

Your help will be much appreciated.

Thanks



Adrian Rosebrock July 27, 2016 at 1:54 am #

Hey Neal — to start, you need to segment the image. Do you have any example images that you're working with?



Neal August 2, 2016 at 8:23 am #

well i'm going to be using different shapes and sizes of objects.



Adrian Rosebrock August 2, 2016 at 10:00 am #

I would use a similar approach to what you're doing with the blocks and place the camera such that it can capture the edge detection to find the blocks. And for labeling the technique to label the shapes.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Anupam September 4, 2016 at 3:40 am #

REPLY ↩

Can you please help me out with detecting overlapping shapes ?



Adrian Rosebrock September 5, 2016 at 8:03 am #

REPLY ↩

For overlapping shapes, I would suggest the [watershed algorithm](#).



Leena October 6, 2016 at 8:25 am #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

How can we use the shapedetector to classify polygon as rectangle/diamond(decision box)/ parallelogram....
please help



Poehe October 31, 2016 at 11:09 am #

REPLY ↩

Hi Adrian, thank you so much for the tutorial, it's a great starting point for me to dive into OpenCV.

I noticed when processing images using your code that in pictures with a white background the engine also shows the contours/edges of the whole input picture as being a shape, while in images with a black background (as in your example) the engine ignores the outside contours of the whole input pic and only shows the contours of the objects within the input pic itself (which is the way it should work, I suppose).

Could you think of a solution that makes the engine n

Your help will be much appreciated!

Cheers



Adrian Rosebrock November 1, 2016 at 11:09 am #

So if I understand your question correctly you want the engine to find shapes themselves? And after thresholding your image to black? I understand that correctly? If so, simply invert the "black" background.



Megha Maheshwari November 21, 2016 at 11:09 am #

Hi Adrian

How can we differentiate between rectangle and trapezoid? A square has equal corners and hence is either a square, rectangle or trapezoid as per my image. However, for square can easily check the width and height, but how do i differentiate between rectangle and trapezoid.



Adrian Rosebrock November 21, 2016 at 12:27 pm #

REPLY ↩

There are many ways to do this. I would consider computing the extent (contour area / bounding box area). A perfect rectangle will have an extent of near 1.0 while trapezoid will have an extent much less than 1.0. You can also compute the angle between each corner of the shape. A rectangle would have near perfect 90 degree angles. Either one will work.



Mohamad November 23, 2016 at 8:12 am #

Hi Great Man. Mr. Adrian

I guess your offer in this tutorial use the webcam for r

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

other shapes such as (H) similar or (L) similar or ... How doing it? Is this method work with edge or point detection?



Adrian Rosebrock November 23, 2016 at 8:30 am #

REPLY ↩

You can certainly use a webcam or Raspberry Pi camera module to perform shape detection. You would just need to read the frame from the camera and process it. I provide tutorials on [how to access webcams here](#).

As for detecting an "H" or "L" you can do that using contour properties (extent, solidity, etc.), template matching, or image descriptors such as Histogram of Oriented Gradients. I would suggest taking a look at [Practical Python and OpenCV](#) along with the [Fundamentals of Recognizing Objects in Images](#).



Luís Serrador November 25, 2016 at 5:48 am #

Hi Adrian,

I tried two of your tutorials (this one and 'OpenCV center of mass') but with the .py file my result is not the same as you show at the end. It only recognizes the first shape/center, and doesn't recognize more shapes.



Adrian Rosebrock November 28, 2016 at 11:34 am #

Hi Luís — what versions of OpenCV and Python are you using?



Luís Serrador November 28, 2016 at 11:34 am #

Hi Adrian! I'm using OpenCV 3.0 and Python 2.7. Any mistake of mine?



Adrian Rosebrock November 28, 2016 at 2:42 pm #

REPLY ↩

Nope, the Python and OpenCV version shouldn't be an issue. I was just curious about the setup to confirm it wasn't an outlier situation. I personally haven't ran into this problem when executing the code. I assume you downloaded the code using the "Downloads" section of this post rather than copying and pasting the code? Sometimes that causes problems with reader's code as well.



Thomas November 30, 2017 at 1:11 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Hi Luis,

you should make an empty file of `__init__.py` , to solve you issue for not detect all the shape from the *.png file.



Jan December 4, 2016 at 1:08 am #

REPLY ↩

Hi Adrian,

Thanks for the tutorial,

I did the same, but for certain circles the vertices were shown as 4 and hence were displayed as squares , can you suggest a way to increase the number of detected vertices in the picture.

Thanks



Adrian Rosebrock December 5, 2016 at 1:08 am #

You'll want to play with the following line

```
approx = cv2.approxPolyDP(c, 0.04
```

The smaller the value passed in for `peri`, the more



Preethi December 23, 2016 at 6:16 am #

Hi Adrian,

From Shape detection i should detect circle . your Same example i need this modification



Adrian Rosebrock December 23, 2016 at 10:50 am #

REPLY ↩

For circle detection, [take a look at this blog post](#). Otherwise, you can use this code to determine a circle as well. A circle will have *many more* approximated contour vertices than all other shapes. You basically need to create an `if` statement for this.



Niel January 5, 2017 at 12:52 am #

REPLY ↩

Hi, Adrian i'm mechatronic students and now i build an obstacle avoidance robot i like to ask how can i create a detection square box in OpenCV for detect colour so it will give feedback to th robot to move or to stop



Adrian Rosebrock January 7, 2017 at 9:00 am #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Hey Niel — can you elaborate on what you mean by a “detection square box”? I’m not sure what you mean.



Chandu January 18, 2017 at 1:23 am #

REPLY ↩

Hi Adrian

i’m getting an error

Traceback (most recent call last):

File “”, line 2, in

from pyimagesearch.shapedetector import ShapeDet

ImportError: No module named pyimagesearch.shapedetector



Adrian Rosebrock January 18, 2017 at 1:23 am #

Hey Chandu — make sure you downloaded the project from the “Downloads” section. It’s likely that your project directory is missing a `__init__.py` file. Please download the project again.



Arturo January 19, 2017 at 9:48 pm #

Hi, i’m having a problem with the code, Some of the image?

I can’t understand the lines 9-12 where you say that v



Adrian Rosebrock January 20, 2017 at 10:57 am #

Hey Arturo, I suggest you [read up on command line arguments](#) before continuing. You don’t need to modify the code at all.



tal January 25, 2017 at 12:14 pm #

REPLY ↩

Hi, i’m having a problem with the code, i have insert the path into the `ap.add_argument("-i", "-image", required=True, help="rec.jpg")` and i’m getting this error.

usage: detect_shapes.py [-h] -i IMAGE

detect_shapes.py: error: argument -i/-image is required.

thanks alot

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Adrian Rosebrock January 26, 2017 at 8:21 am #

REPLY ↩

You **do not** have to modify the code at all. You just need to supply the `--image` switch to the Python script via command line argument:

```
$ python detect_shapes.py --image shapes_and_colors.png
```

Please read up on [command line arguments](#) before continuing.



BKumar March 2, 2017 at 2:07 am #

REPLY ↩

Hey Adrian,

I was wondering about how to find out the number of squares, 2 rectangles etc., how can you label them in the image and display the number of instances of each shape in the image?

Thanks in advance.



Adrian Rosebrock March 2, 2017 at 6:40 am #

I would use a Python built-in dictionary to store the counts. I'll loop over them. Your pseudocode might look something like this:

```
1 shapes = {}
2 for shape in detectedShapes:
3     shapes[shape] = shapes.get(shape, 0) + 1
```

This would give you a dictionary of shape counts like this:

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Milán Vincze March 5, 2017 at 1:42 pm #

REPLY ↩

Hello!

I get an error when I run the program:

ImportError: No module named 'imutils'

I installed imutils before. Could you please help?



Adrian Rosebrock March 6, 2017 at 3:42 pm #

REPLY ↩

The first part of this blog post discusses exactly how to solve this error message:

```
$ pip install imutils
```

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Milán Vincze March 6, 2017 at 4:42 pm #

REPLY ↩

Yes I know I ran that before I start the program. Maybe the problem is that the raspberry want to run it in python3. How can I run in python2 if I installed python2 and python 3 too? I thought that is the problem because the installation put the imutils in python2.7 library.



Milán Vincze March 6, 2017 at 4:46 pm #

REPLY ↩

I also copied the imutils folder to python3.1 folder



Milán Vincze March 6, 2017 at 4:46 pm #

Sorry, I copied the imutils to python3.1 folder. I thought it would be better to bother you.



Adrian Rosebrock March 8, 2017 at 12:15 pm #

For what it's worth, you could also try:

```
$ pip3 install imutils
```



Milán Vincze March 12, 2017 at 6:46 am #

Hello! Adrian

First thank you for the excellent tutorial! I am interesting in it because there is a tutorial for it that you made?

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Adrian Rosebrock March 13, 2017 at 12:15 pm #

REPLY ↩

I would suggest using the `cv2.VideoCapture` class or the `VideoStream` class covered in [this blog post](#). My book, [Practical Python and OpenCV](#) also covers the basics of working with video streams and video files — this would help you port the code over to a video stream rather than a single image.



Tyler April 16, 2017 at 1:13 pm #

REPLY ↩

Adrian can u suggest what change I should make to the code to run in real time using laptop camera feed ,I would be very grateful if you can help me.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

**Adrian Rosebrock** April 19, 2017 at 1:06 pm #

REPLY ↩

The comment you replied to has a link to a blog post + book that I recommend that you read so you can access your laptop webcam. Take the time to study the basics of OpenCV first, then it will be easy to implement this method for real-time applications.

**PANJI** March 14, 2017 at 3:48 pm #

REPLY ↩

Sir i have problem error

```
usage: detect_shapes.py [-h] -i IMAGE
```

```
detect_shapes.py: error: argument -i/-image is required
```

anyone help me?

**Adrian Rosebrock** March 15, 2017 at 8:00 am #

Please read the comments before you post. See the comment by "Arturo" above.

**Open The CV** March 19, 2017 at 11:38 pm #

Hi, how can I find the angle between a triangle and a line?

**Adrian Rosebrock** March 21, 2017 at 7:00 am #

I'm not sure what you mean. Do you have an example image of what you're working with?

**Open The CV** March 21, 2017 at 7:36 am #

REPLY ↩

Problem solved with some trigonometry formulas. Thanks

**sandra** May 8, 2017 at 12:44 pm #

REPLY ↩

Hello, how can i detect certain rectangle which is the border of some text area in an image ?

**Adrian Rosebrock** May 8, 2017 at 12:44 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Hi Sandra — I'm not sure I understand your question. It would be helpful if you had an example image of what you're working with.



Abu May 11, 2017 at 6:51 pm #

REPLY ↩

Hello Adrain,

Awesome tutorial, thank you.

I was wondering if there is a way to detect rectangles or squares to exact approximately. Basically, I have an image, and it has shapes and text in it. I only want to detect the shapes and ignore all the text in the image. This tutorial really helped me. But, I am still detecting shapes. Is there a way, I can completely ignore that?.

Thanks.



Adrian Rosebrock May 15, 2017 at 9:00 am #

I would compute the solidity of the shape and compare it to the convex hull area. Text will have a lower solidity than shapes.



Laura May 30, 2017 at 2:46 pm #

Hello Adrian,

Thanks a lot for the awesome tutorial.

I am using the shape detection to get the coordinates of the shapes in the image that I am getting from my phone camera. For some reason, when I start to place elements in the same row, the coordinates are in a different order.

However, when I try the same layout from an image I am building in photoshop (not a picture) it works correctly. When I compared both shape detection processes, I realised that the order in which the shapes are sorted is also changing. I tried altering the lightning and contrast of the pictures, but somehow there is something that is altering the order of how the elements are detected and the order of how the coordinates are organised.

Image from camera: <http://imgur.com/a/xHaGn>

Digital image: <http://imgur.com/a/NNsv5>

Do you have any idea of what could be the problem?

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Adrian Rosebrock May 31, 2017 at 1:00 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

The `cv2.findContours` function will not return contours in a specific order. You should [sort your contours](#) if you expect them to be in a given order.



Martin June 14, 2017 at 10:21 am #

REPLY ↩

Hi Laura,

I am working on the same topic -> finding squares on picture taken by my phone camera.
Could you share your project with us?
Thanks



jandi May 30, 2017 at 7:43 pm #

Thanks for this tutorial ...

I have a question. In my image, I have a square and I want to know how can I do that?



Adrian Rosebrock May 31, 2017 at 1:00 pm #

I would suggest using either:

1. Contour properties, such as extent, solidity, and area
2. Features, such as Hu Moments or Zernike Moments



Thailynn June 19, 2017 at 12:35 pm #

Hi Adrian,

Is there a way to save the results of the image classification into a new image? For example, identifying all the squares and saving a new image with just the squares. I am trying to do some feature extraction of satellite imagery.

Thank you very much for any insight you can offer!



Adrian Rosebrock June 20, 2017 at 10:55 am #

REPLY ↩

Hi Thailynn — if you are trying to extract each individual square, I would compute the bounding box and extract it using array slicing. You can then write the region to disk using the `cv2.imwrite` function. I cover this in more detail inside [Practical Python and OpenCV](#).



Cuningan July 23, 2017 at 12:55 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Hello Adrian, i ahve bought you curse and now i am seeking this tutorial.
I am using RPI with with OpenCV3.2 compiled like you described on you post.

My problem is that also, python2 and 3 can not find imutils, i have tryed to install it with pip but failed with a large crash...

Any idea about??



Adrian Rosebrock July 24, 2017 at 3:33 pm #

REPLY ↩

Which tutorial did you use to install OpenCV? Also, what is the error you are getting when trying to install imutils? Without knowing the error, I can't



disheet August 8, 2017 at 3:46 am #

Hello Sir,

I am getting below error.Please give me the solution

File "/usr/local/lib/python2.7/dist-packages/imutils/opencv2.py", line 10, in
(h, w) = image.shape[:2]

AttributeError: 'NoneType' object has no attribute 'shape'



disheet August 8, 2017 at 3:48 am #

And i am using logitech c170 webcam



Adrian Rosebrock August 10, 2017 at 8:00 am #

It sounds like OpenCV cannot access your webcam. I discuss NoneType errors (and how to resolve them) in detail inside [this post](#).



disheet August 11, 2017 at 1:57 am #

REPLY ↩

Hello Sir,

How can i do shape detection using webcam?



Adrian Rosebrock August 14, 2017 at 1:00 am #

REPLY ↩

You apply the exact same algorithm detect faces. You can access the frames of a video using [this post](#)

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



morejump September 10, 2017 at 12:49 am #

REPLY ↩

Hi Adrian,

the rectangle is shape which has 4 vertices, and also has three of them is 90 degree. cause your algorithm dose not work every time



Doson September 26, 2017 at 4:29 am #

REPLY ↩

I don't think it is strict to recognize the sharp rhomboid have for 'approx' but it is neither a square c checking the degree in each 'approx' .If more than th believe it is a square or rectangle.



David October 11, 2017 at 5:08 am #

Just wanted to say that these tutorials have



Adrian Rosebrock October 13, 2017 at

Thanks David, I really appreciate that 😊



KansaiRobot October 23, 2017 at 5:08 am #

Very nice Blog, I have just discovered it. I am going to be learning through it soon. I would like to ask you a question. In this post you covered shape detection when the objects are quite separated from each other- therefore you can find the true center.

How about detecting shapes when one shape is touching another or even slightly overlapping it?



Adrian Rosebrock October 23, 2017 at 6:10 am #

REPLY ↩

Take a look at the [Watershed algorithm](#).



Enko October 27, 2017 at 5:56 am #

REPLY ↩

Hi Adrian,

Thank's for your excellent tutorial, I want to detect circle detection part:

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

```
else:  
    # shape = "circle"  
    img = np.zeros(image.shape, image.dtype)  
    cv2.drawContours(img, [c], -1, (255, 255, 255), -1)  
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
    circles = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1.2, 100)  
    if circles is not None:  
        shape = "circle"  
    but the "circles" is always None. Can you help me?
```



Adrian Rosebrock October 27, 2017 at

Hi Enko — see this blog post: [Detecting](#)



KansaiRobot October 29, 2017 at 8:15 pm #

Part 1) I would like to ask if watershed can be used to detect circles. In particular tablet-like. (I tried the method and



Adrian Rosebrock October 31, 2017 at

In general, yes, the watershed algorithm can obtain a reasonable segmentation.



KansaiRobot October 31, 2017 at 8:

Yes, segmentation is not a problem. I used a discriminant analysis method for segmentation. I can send you the original (non segmented) images or the binarized ones (just purely black and white)



KansaiRobot October 29, 2017 at 8:16 pm #

An example of my problems can be found here
<https://stackoverflow.com/questions/46107628/problems-with-segmenting-pills>
I would like to hear your advice on how to tackle this problem. Thanks in advance



Adrian Rosebrock October 31, 2017 at

Do you have the original example image



Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

REPLY ↩

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



KansaiRobot October 31, 2017 at 8:28 pm #

REPLY ↩

Thanks. I have some sample images obtained with backward illumination. (Therefore basically dark shapes on white background). I can send some to you for the post you are considering.



Adrian Rosebrock November 2, 2017 at 2:36 pm #

REPLY ↩

Sure, that would be cool to take. If you provide me with your email address you can send the images to me.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



KansaiRobot October 29, 2017 at 8:17 pm #

Part2) You seem to be using scikit watershed



Adrian Rosebrock October 31, 2017 at 8:17 pm #

At the time the [watershed](#) blog post was written, the watershed analysis functions in OpenCV were not as easy to use as they are now and I would recommend using whichever one you prefer.



Kanwal November 21, 2017 at 9:14 am #

Hello Sir,

Can you help how to solve the problem of occlusion.



Sameer November 26, 2017 at 5:14 am #

REPLY ↩

Hi Adrian,

When I run the program, only single object is being contoured. What is wrong?



Adrian Rosebrock November 27, 2017 at 1:09 pm #

REPLY ↩

Make sure you click the active window where you are running the detection.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Yadnyesh December 1, 2017 at 10:32 am #

REPLY ↩

Hello Adrian,

I have slight problem

The code works fine with images which have a black background it detects all the shapes
the problem comes when an image has a white background.

It recognizes the frame of image as a rectangle and none of the shapes within it.

Can you help me with this?

Thank you in advance



Yadnyesh December 1, 2017 at 11:31 am #

Sorry for this I read your reply for a com

So all I have to do this

replace

`thresh = cv2.threshold(blurred, 60, 255, cv2.TH`
with

`thresh = cv2.threshold(blurred, 255, 60, cv2.TH`

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Priyam Vora December 20, 2017 at 2:30 am #

Hi Ardrian,

How can we detect a cricket bat in a image? Please

Thank You



Caner Yagci December 21, 2017 at 11:12 am #

Hi Adrian, What should I do if I want to detect only the circles ?

Thank you



Adrian Rosebrock December 22, 2017 at 6:49 am #

REPLY ↩

If you want to detect only the circles you should take a look at [this blog post](#) where I filter contours or apply the [Hough Circles method](#).



james December 23, 2017 at 12:50 pm #

REPLY ↩

hello can you help me to detect an custom
that with countours or something else

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Adrian Rosebrock December 26, 2017 at 4:31 pm #

REPLY ↩

Hey James — do you have any example images of what you're working with?



Emre uysal December 24, 2017 at 9:12 am #

REPLY ↩

hi adrian

how can i detect resistor shape or undefined shape?
thanks



Adrian Rosebrock December 26, 2017

I'm not sure what you mean by "undefined shape". The Histogram of Oriented Gradients method is used for shape detection.



Oshada January 8, 2018 at 7:33 am #

How to apply this technique for unfilled shapes?



Adrian Rosebrock January 8, 2018 at 2:00 pm #

By "unfilled" do you mean just the outline of the shape?

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Oshada January 9, 2018 at 10:31 am #

REPLY ↩

Yeah. It works now. But, the problem is I have to change the threshold when a different image is used. I want to convert a hand-drawn flow chart into a stored procedure. How do I solve this issue?



Adrian Rosebrock January 10, 2018 at 12:56 pm #

REPLY ↩

Take a look at "adaptive thresholding". Adaptive thresholding works by thresholding on local areas. It might work well for you in this situation. Also take a look at Otsu's thresholding. I cover both inside [Practical Python](#) and [OpenCV](#).

Mustafa Ismail January 9, 2018 at 4:10 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Hello, thank you for the awesome tutorial man!

I'm tryna detect a shape like this <https://drive.google.com/open?id=1D4SiYQ84TYDA0aZxaEDm5MDIY6IWWaW>

and as you can see the shape on the right is perfectly detected by following your tutorial and tuning some parameters. However i'm having a problem on detecting half colored shapes like the shape on the left, as you can see the thresholding technique classifies the blue part as black.

So any ideas on how to solve this? I need to detect the whole object as a triangle



Adrian Rosebrock January 10, 2018 at 12:50 pm #

REPLY ↩

You will need to play with the thresholding (or edge detection instead. Secondly, make sure the shape you want to detect) and the background. I detection methods will not work.



Oshada February 1, 2018 at 2:25 am #

Hi Adrian, I tried to identify shapes in hand-drawn

<https://ibb.co/d10sEG>

The case is sometimes because these are hand-drawn (corners) which mess up the detection. So I used close gap between shapes and arrows and text is less, even contours. Is there are way to get rid of this problem with this much of gap between items. Also, I can't get rid of

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Adrian Rosebrock February 3, 2018 at

There are a few ways to approach this problem, but have you considered training an object detector to recognize each of these shapes? [HOG + Linear SVM](#) would be a good starting point. This would alleviate the need to rely on image processing operations and make your algorithm more extendible.



Oshada February 3, 2018 at 8:19 pm #

REPLY ↩

The problem is my supervisor has highly recommended me not to use any ML or DL approaches first. That is why I have been trying to use contour analysis, mathematical relationships among different shapes and etc. I have done detection parts and everything putting that restriction I mentioned. And I want to get rid of it. Can you give me some tips to do that without going for ML.

Adrian Rosebrock February 6, 2018 at

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



You'll need to continue to fiddle with the parameters to your morphological operations if you want to continue using non-ML solutions. That's perfectly fine from an education standpoint but keep in mind that your method won't be as robust.



Juhi March 5, 2018 at 12:35 pm #

REPLY ↩

ModuleNotFoundError: No module named 'pyimagesearch'

Please help.



Adrian Rosebrock March 7, 2018 at 9:20 am #

You need to use the "Downloads" section of the [OpenCV](#) archive. Change directory to where the code is stored.



Darshan kumar March 17, 2018 at 3:35 am #

Hi Adrian! I am looking for a code which has a function to find the co ordinates on a plane. And may be midpoint.



Andrew R March 18, 2018 at 9:58 am #

Hi Adrian

Thanks for posting this – and your other examples – really helpful.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Adrian Rosebrock March 19, 2018 at 5:12 pm #

REPLY ↩

Thanks Andrew, I'm glad you enjoyed the post! 😊



Tom S March 22, 2018 at 10:18 pm #

REPLY ↩

Hi Adrian,

Thanks for this and all the other awesome posts!

I'm really new to OpenCV and have been trying recently to implement it into my capstone project for undergrad. I'm using a number of your tutorials for help in this. The problem I'm running into with this one is identifying shapes that are really close together. I need a way to separate them so I can translate it to code using a syntax my team can understand.

<https://imgur.com/a/OjwE5>

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

So far it's only detecting the entire block structure like so:

<https://imgur.com/a/8cvRP>

Any help you could offer in finding shapes that are closer together would be greatly appreciated. Thank you in advance!

Best,
Tom



Adrian Rosebrock March 27, 2018 at 6:42 am #

REPLY ↩

Keep in mind that you'll only be able to clearly segment each piece via thresholding, edge detection, etc. You can't find all the legos at once. It's impossible for the cv2.findContours to find each individual lego. Instead, I would recommend you to find each set of legos.



Tom S March 27, 2018 at 2:14 pm #

Ignore the previous comment, I did shape detection. It worked like a charm, thank



Adrian Rosebrock March 30, 2018 at 10:00 am #

Awesome! Congrats, Tom 😊



Tom S March 27, 2018 at 2:12 pm #

REPLY ↩

I decided to detect the shapes after each color had been detected and that worked well!



Kay March 28, 2018 at 3:03 am #

REPLY ↩

Hi Adrian,

I have two rectangles to be detected but it only detects one and from the previous reply of yours, I need to press any key on the keyboard but it closed the program and does not advance the detection. How to make them show all the detected shape in one go?



Adrian Rosebrock March 30, 2018 at 7:00 am #

It sounds like the second rectangle is:

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

1. Not being detected by the threshold process
2. If it is being detected, the contour approximation is not returning 4 coordinates

You will need to debug both of these. I would suggest starting with the "thresh" map (you can debug it using `cv2.imshow()`).

Additionally, if you are new to working with OpenCV and Python I would recommend reading through [Practical Python and OpenCV](#) to help you get up to speed. I hope that helps point you in the right direction!



kasun April 2, 2018 at 8:59 pm #

REPLY ↩

i used this approach to identify the er digram
identify the double rectangle (Week entity)double elipse
)can you dive mee tip for identiy those shapes



kasun April 2, 2018 at 9:11 pm #

those er digrams are hand drawn ones so u
can do this easily using ml or dl .plese tell me easyes



Adrian Rosebrock April 4, 2018 at 12:11 pm #

Do you have any example images of the



kasun April 5, 2018 at 9:01 pm #

ya i have .this is link for that
https://drive.google.com/open?id=1JEJ_RySZ04FYIVYI3Zf1jjryAR5OmYPU
only there you can see double rectangle and double diamond also have double elipse and doted elipse too.



Adrian Rosebrock April 6, 2018 at 8:47 am #

REPLY ↩

There are a few things you can do for this. You either extract the contour hierarchy for the double rectangle/double diamond or you can extract the outer shape and then do another series of contour detections to detect any inner shapes.



Manuel April 12, 2018 at 10:59 pm #

works great ! but with 200x150 resolution n
can i do ? Thank you !

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Adrian Rosebrock April 13, 2018 at 6:39 am #

REPLY ↩

Can you increase the resolution of your input images?



Mitch April 24, 2018 at 4:18 pm #

REPLY ↩

Thanks for the great tutorials. Can you explain why you initially resize the image ("load the image and resize it to a smaller factor so that the shapes can be approximated better") and its implication about improved performance? Thank you.



Adrian Rosebrock April 25, 2018 at 5:31 am #

While high resolution images are appealing, they can be a problem for many computer vision algorithms. The larger an image is, the slower an algorithm will run. Secondly, reducing an image's resolution, which can be seen as a "noise reduction", allows you to focus on what actually matters.

I hope that helps and if you're interested in learning more about computer vision be sure to refer to [Practical Deep Learning for Python](#) you get up to speed quickly.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



ali May 14, 2018 at 5:31 am #

Hi Adrian

i have a image with green bounding box, i want to get coordinates of bounding box. please guide me.



Adrian Rosebrock May 14, 2018 at 11:47 am #

REPLY ↩

I'm a little confused here. Typically we create computer vision algorithms to detect the (x, y)-coordinates of an object in an image and then draw a bounding box surrounding it. It sounds like you have the opposite workflow — you have a green bounding box already in an image and you want to detect the location of it?



ali May 15, 2018 at 12:20 am #

REPLY ↩

Yes, precisely, I want to get the position of the object in the same place.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Adrian Rosebrock May 17, 2018 at 7:12 am #

REPLY ↩

Got it, I understand now. There are two options here.

1. Apply [color thresholding](#) to find green regions. Then process the contours to filter out only the box itself.
2. [Train an object detector](#) to find only squares.

The second option would be more accurate but would require significantly more work.



Yadira May 22, 2018 at 2:10 pm #

Hi Adrian,

When I try to run the code I keep getting "the following
it may have the simplest solution in the world, but I'm



Adrian Rosebrock May 23, 2018 at 7:1

Reading [this post on command line arg](#)



Dan June 2, 2018 at 2:39 pm #

Dear Adrian,

Awesome work! It is quite useful.
I have a question regarding the edge effect.

The contour approximation goes wrong when I tested
Does this mean that the method works perfectly, assuming the edge is smooth?

I wonder whether you have published a paper about this?
I'm not a computer vision newbie and many thanks in advance for your explanation.

cheers,



Dan June 2, 2018 at 2:40 pm #

REPLY ↩

I mean I am a newbie in computer vision ahh



Adrian Rosebrock June 5, 2018 at

Yes, this method does assume that
background. If you cannot obtain a clean seg

DEDIV ↩

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Additionally, if you are new to OpenCV/computer vision I would definitely recommend working through my book, [Practical Python and OpenCV](#). I wrote this book for readers new to computer vision as you are. Give it a try, I'm confident that once you go through it you'll be able to more easily understand the concepts discussed on this blog.



Dan June 2, 2018 at 3:27 pm #

REPLY ↩

Hi Adrian,

Can this method detect other shapes such as an octagon, hexagon, etc?

Many thanks!



Adrian Rosebrock June 5, 2018 at 8:02 pm #

Yes, you would just need to check the `cv2.isContourConvex()` (I have done in this example).



Geetanjali Sahoo June 13, 2018 at 2:23 am #

hi Adrian,

how to know the connector between two shape ? like



Adrian Rosebrock June 13, 2018 at 5:21 pm #

There are a few ways to do this, some use the `cv2.contourArea()` contour approximation technique discussed here or you could compute the solidity and extent of the contour and use them to help determine if a particular shape is an arrow.



Henrique July 21, 2018 at 10:46 pm #

REPLY ↩

Hi Adrian !

I'm working on computer vision applied to drones, and I need to calculate the area of a shape on the ground, seen from a camera on the copter. I've seen this tutorial and "Measuring size of objects in an image" also. Is there any implementation in which both could return the area of the shape ? Thank you, appreciate your work a lot !



Adrian Rosebrock July 25, 2018 at 8:21 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Hey Henrique — your project is a very specific and you'll need more advanced techniques here. I would suggest first calibrating the camera via an intrinsic/extrinsic calibration computation. From there you'll be able to compute a more accurate measurement of the area.



Andrea September 12, 2018 at 2:41 pm #

REPLY ↩

Adrian, please could you tell us where we can see how the function `imutils.resize()` is defined?

Thanks



Adrian Rosebrock September 14, 2018 at 1:11 pm #

The "imutils.resize" function is defined in `imutils.py`.



Adnan KHan October 15, 2018 at 4:15 pm #

Hi,

I want to extract the text (in order) from an image using `pytesseract`. I want to find the text in the image first then split the image based on those rectangles using `pytesseract`.

So how would I detect the rectangles in this image? I have attached the image.



Adrian Rosebrock October 16, 2018 at 1:11 pm #

I would first perform a perspective transform on the image to get a top-down view of your paper. From there, a Hough lines transform would be used to find the lines in the image.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



Josh Doran October 22, 2018 at 12:29 pm #

REPLY ↩

When running the code, only the pentagon is recognized, none of the other shapes are looped through. Can't figure out why this happens.



Adrian Rosebrock October 22, 2018 at 1:02 pm #

REPLY ↩

You need to click on the active window opened by OpenCV and press any key on your keyboard to advance execution of the script.



pavan varyani November 6, 2018 at 2:40 pm #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

hello Adrian, thanks for the lovely tutorials.

i have an issue, the green contours are not being drawn at the right location , they are being drawn at random locations,rest (centre detection,shape detection seemd to be perfect).



Adrian Rosebrock November 10, 2018 at 10:22 am #

REPLY ↩

That's quite odd. Are you using the code and images from this blog post? Or your own custom images?



pavan varyani November 12, 2018 at 7:07 am #

Hey the code works fine now, i just changed the amount of the width 😊



pavan varyani November 9, 2018 at 2:01 am #

Hello Adrian thanks for posting such extraor
i have a very fundamental doubt, could you please er
we begin with the manipulations.



Adrian Rosebrock November 10, 2018 at 10:22 am #

1. The less data we have to process, the better the performance.
2. While higher resolution images may look aesthetically pleasing, they can be a performance hit. We reduce image size as a form of optimization.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE



pavan varyani November 12, 2018 at 7:07 am #

REPLY ↩

thanks 😊 ,could you please brief me on the approach to the picture containing shapes (drawn on paint tool) ? the same code when tried on a customised picture containing shapes (made on paint tool), detects only the shape of the canvas part of the picture.



Nirmal Mandal November 20, 2018 at 6:31 am #

REPLY ↩

Hello, can u tell me how can detect hologram from national id card.

Adrian Rosebrock November 20, 2018 at 6:31 am #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Sorry, I do not have any experience with hologram detection.



Arya November 28, 2018 at 1:36 am #

REPLY ↩

Sir please help me install open CV in my laptop to start doing my project.
My system is operating in windows.
Please reply me



Adrian Rosebrock November 30, 2018

Hi Arya — I have a number of [OpenCV](#) support macOS and Linux here on the PyImageSearch blog. Do you use macOS or Linux for computer vision.



boba December 11, 2018 at 3:51 pm #

Adrian,I am working on my school project where I would like to store the location and co-ordinates of the object (python)?



Adrian Rosebrock December 13, 2018

What specific coordinates? The bounding box? The outline/contour of the shape? Just the center (x, y)?



boba December 15, 2018 at 8:43 am #

REPLY ↩

Adrian,

Automatically detect/identify different possible shapes (coordinates of the outline/contour of those shapes) from any given image (things/people/objects).



Adrian Rosebrock December 18, 2018 at 9:18 am #

REPLY ↩

You would need to train your own custom object detector. Be sure to refer to [this guide](#) for more information.



Amvoxite December 13, 2018 at 1:17 am #

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Adrien,

What do you make of this problem finding `imutils grab_contours(...)`?

When I run this script I get:

`AttributeError: 'module' object has no attribute 'grab_contours'`



Adrian Rosebrock December 13, 2018 at 8:55 am #

REPLY ↩

You need to upgrade your `imutils` install:

```
$ pip install --upgrade imutils
```



John Kent December 31, 2018 at 2:30 pm #

Long thin rectangle (14:1) being detected as

Tried every combination of modes & methods.

Happy to send code and test image.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

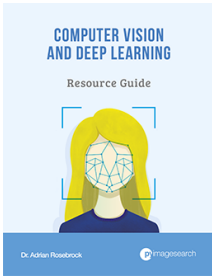
SUBMIT COMMENT

Search...



Resource Guide (it's totally free).

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



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.

[Download for Free!](#)

Deep Learning for Computer Vision with Python Book — OUT NOW!



You're interested in deep learning and computer vision, *but you don't know where to start*. I will teach you all you need to know about deep learning.

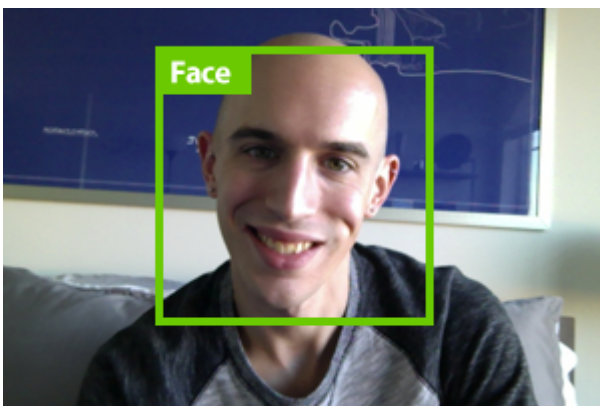
[CLICK HERE TO MASTER DEEP LEARNING](#)

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

[START MY EMAIL COURSE](#)

You can detect faces in images & video.



Are you interested in **detecting faces in images & video**? Let me help! I guarantee that my new book will turn you into a face detection expert. Give it a shot yourself.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

[CLICK HERE TO MASTER FACE DETECTION](#)

PyImageSearch Gurus: NOW ENROLLING!

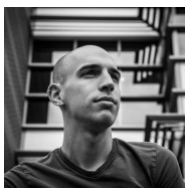
The PyImageSearch Gurus course is *now enrolling!* In

- Automatic License Plate Recognition (ANPR)
- Deep Learning
- Face Recognition
- *and much more!*

Click the button below to learn more about the course

TAKE A TOUR & GET 10 (FREE) LESSONS

Hello! I'm Adrian Rosebrock.



I'm an entrepreneur and Ph.D who has launched two successful image search engines, [ID My Pic](#) and [Chic Engine](#). I'm here to share my tips, tricks, and hacks I've learned along the way.

Learn computer vision in a single weekend.

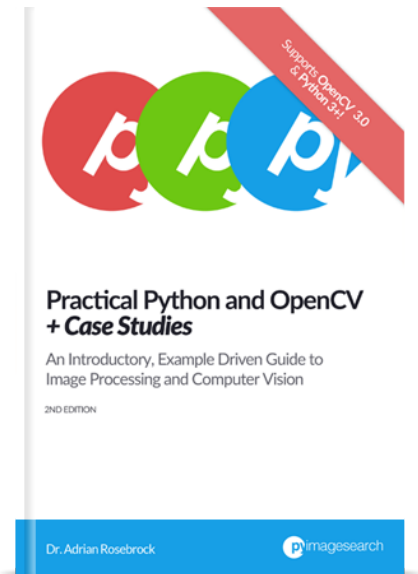
Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning



Want to learn computer vision & OpenCV? I can teach you in my new book is your **guaranteed, quick-start guide** to become a computer vision ninja.

[CLICK HERE TO BECOME AN OPENCV NINJA](#)

Subscribe via RSS



Never miss a post! Subscribe to the PyImageSearch engine tutorials, tips, and tricks

POPULAR

Install guide: Raspberry Pi 3 + Raspbian Jessie + OpenCV
APRIL 18, 2016

Raspbian Stretch: Install OpenCV 3 + Python on your Raspberry Pi
SEPTEMBER 4, 2017

Home surveillance and motion detection with the Raspberry Pi, Python, OpenCV, and Dropbox
JUNE 1, 2015

Install OpenCV and Python on your Raspberry Pi 2 and B+
FEBRUARY 23, 2015

Ubuntu 16.04: How to install OpenCV
OCTOBER 24, 2016

Basic motion detection and tracking with Python and OpenCV
MAY 25, 2015

How to install OpenCV 3 on Raspbian Jessie
OCTOBER 26, 2015

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Interested in computer vision, OpenCV, and deep learning, but don't know where to start? Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning

Find me on [Twitter](#), [Facebook](#), [Google+](#), and [LinkedIn](#).

© 2019 PyImageSearch. All Rights Reserved.

Free 17-day crash course on Computer Vision, OpenCV, and Deep Learning



Interested in computer vision, OpenCV, and deep learning, but don't know where to start?

Let me help. I've created a *free*, 17-day crash course that is hand-tailored to give you the best possible introduction to computer vision and deep learning. Sound good? Enter your email below to get started.

START MY EMAIL COURSE

**Free 17-day crash course on
Computer Vision, OpenCV, and Deep
Learning**