

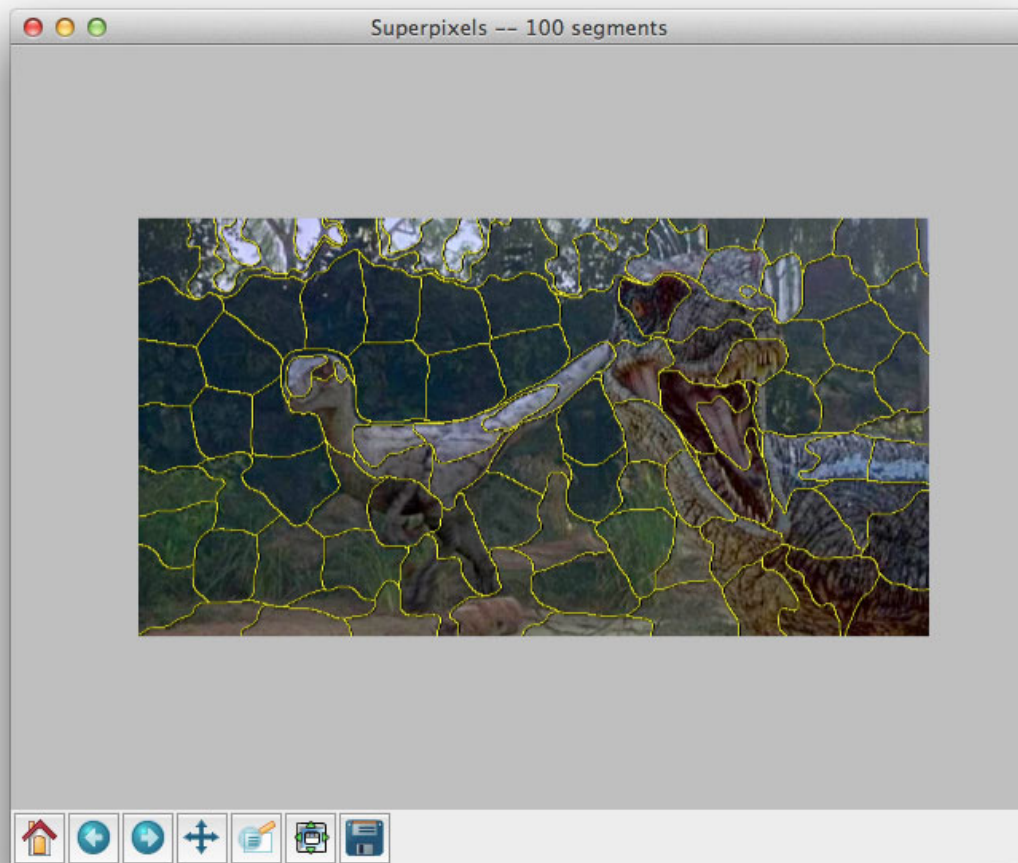
PYIMAGESEARCH

[IMAGE PROCESSING \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/IMAGE-PROCESSING/\)](https://www.pyimagesearch.com/category/image-processing/)

[TUTORIALS \(HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/TUTORIALS/\)](https://www.pyimagesearch.com/category/tutorials/)

Segmentation: A SLIC Superpixel Tutorial using Python

by [Adrian R](#)



https://pyimagesearch.com/wp-content/uploads/2014/07/slic_100_segments.jpg

Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Imagine, I'm not exactly happy.

However, to combat this persistent, often distracting nuisance, I've started sleeping more regularly, drinking substantially less caffeine (I was only drinking two cups of coffee a day to begin with), and consuming next to no alcohol. Which is a real challenge when you enjoy a frothy beer as much as I do.

More sleep has certainly helped. And I don't miss the occasional beer.

But man, these caffeine headaches **are the worst**.

Luckily, I woke up this morning to an email that practically cured me of my caffeine free woes and interminable eyelid twitching.

Opening my inbox, I found an excellent question from dedicated PyImageSearch reader Sriharsha Annamaneni, who wrote in and asked:

“Hi Adrian,

In one of **Jitendra Malik's YouTube videos (<https://www.youtube.com/watch?v=Mqg6eorYRIQ>)**, he explains superpixels. I did not understand it completely. Can you explain it on your website?

Thank you,

Sriharsha Annamaneni

Great question Sriharsha. I'd be happy to discuss superpixels, I'm sure a ton of other PyImageSearch readers would be interested too.

But before we dive into some code, let's discuss what exactly a superpixel is and why it has important applications in the computer vision domain.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Looking for the source code to this post?

[JUMP RIGHT TO THE DOWNLOADS SECTION](#) →

OpenCV and Python versions:

This example will run on **Python 2.7/Python 3.4+** and **OpenCV 2.4.X/OpenCV 3.0+**.

So, What's a Superpixel?

Take a second, clear your mind, and consider how we represent images.

Images are represented as a grid of pixels, in either single or multiple channels.

We take these $M \times N$ pixel grids and then apply algorithms to them, such as face detection and recognition, template matching, and even deep learning applied directly to the raw pixel intensities.

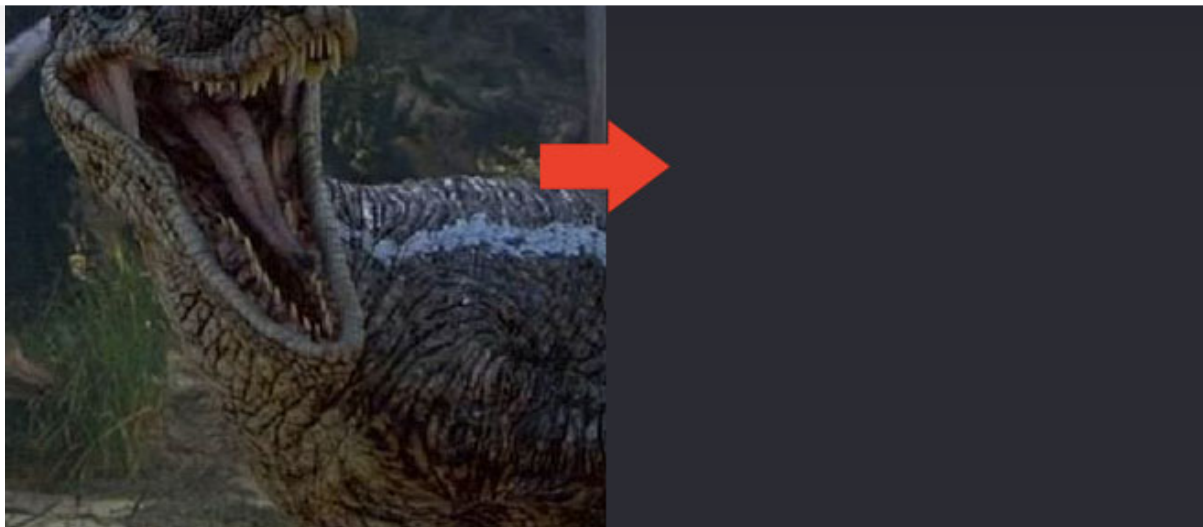
The problem is that our pixel grid is ***by no means*** a natural representation of an image.

For example, consider the image below:



👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer



https://pyimagesearch.com/wp-content/uploads/2014/07/slic_example_pixel.jpg

Figure 1: If I were to take the *single pixel* pointed to by the red arrow (*left*) and show it to you on the *right* at an obviously zoomed-in scale, would you be able to tell me anything about the image based solely on that pixel? Anything descriptive? Or of semantic value?

If I were to take a single pixel from the image on the *left* (highlighted by the red arrow) and then showed it to you on the *right*, could you reasonably tell me that (1) that pixel came from a Velociraptor and (2) that single pixel actually holds some sort of semantic meaning?

Unless you really, *really* want to play devils advocate, the answer is a resounding no — a single pixel, standing alone by itself, is not a natural representation of an image.

Taken as a whole, this grouping of pixels as a pixel grid is simply an artifact of an image — part of the process of capturing and creating digital images.

That leads us to superpixels.

Intuitively, it would make more sense to explore not only perceptual, but semantic meanings of an image formed by locally grouping pixels as well.

When we perform this type of local grouping of pixels on our pixel grid, we arrive at ***superpixels***.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

To summarize the work of [Dr. Xiaofeng Ren \(http://ttic.uchicago.edu/~xren/research/superpixel/\)](http://ttic.uchicago.edu/~xren/research/superpixel/), we can see that a mapping from pixel grids to superpixels would (ideally) hold the desirable properties of:

- 1 **Computational efficiency:** While it may be (somewhat) computationally expensive to compute the actual superpixel groupings, it allows us to reduce the complexity of the images themselves from hundreds of thousands of pixels to only a few hundred superpixels. Each of these superpixels will then contain some sort of perceptual, and ideally, semantic value.
- 2 **Perceptual meaningfulness:** Instead of only examining a single pixel in a pixel grid, which carries very little perceptual meaning, pixels that belong to a superpixel group share some sort of commonality, such as similar color or texture distribution.
- 3 **Oversegmentation:** Most superpixel algorithms *oversegment* the image. This means that most of important boundaries in the image are found; however, at the expense of generating many insignificant boundaries. While this may sound like a problem or a deterrent to using super pixels, it's actually a positive — the end product of this oversegmentation is that that very little (or no) pixels are lost from the pixel grid to superpixel mapping.
- 4 **Graphs over superpixels:** Dr. Ren refers to this concept as “representationally efficient”. To make this more concrete, imagine constructing a graph over a $50,000 \times 50,000$ pixel grid, where each pixel represents a node in the graph — that leads to very large representation. However, suppose we instead applied superpixels to the pixel grid space, leaving us with a (arbitrary) 200 superpixels. In this representation, constructing a graph over the 200 superpixels is substantially more efficient.

Example: Simple Linear Iterative Clustering (SLIC)

As always, a PyImageSearch blog post wouldn't be complete without an example and some code.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

 → [Click here to download the code](#)

SLIC Superpixel Segmentation in Python and scikit-image


```

1. | # import the necessary packages
2. | from skimage.segmentation import slic
3. | from skimage.segmentation import mark_boundaries
4. | from skimage.util import img_as_float
5. | from skimage import io
6. | import matplotlib.pyplot as plt
7. | import argparse
8. |
9. | # construct the argument parser and parse the arguments
10. | ap = argparse.ArgumentParser()
11. | ap.add_argument("-i", "--image", required = True, help = "Path to the image")
12. | args = vars(ap.parse_args())
13. |
14. | # load the image and convert it to a floating point data type
15. | image = img_as_float(io.imread(args["image"]))
16. |
17. | # loop over the number of segments
18. | for numSegments in (100, 200, 300):
19. |     # apply SLIC and extract (approximately) the supplied number
20. |     # of segments
21. |     segments = slic(image, n_segments = numSegments, sigma = 5)
22. |
23. |     # show the output of SLIC
24. |     fig = plt.figure("Superpixels -- %d segments" % (numSegments))
25. |     ax = fig.add_subplot(1, 1, 1)
26. |     ax.imshow(mark_boundaries(image, segments))
27. |     plt.axis("off")
28. |
29. | # show the plots
30. | plt.show()

```

On **Lines 2-7** we import the packages we'll be using for this example. To perform the SLIC superpixel segmentation, we will be using the [sckit-image implementation \(http://scikit-image.org/docs/dev/api/skimage.segmentation.html#skimage.segmentation.slic\)](http://scikit-image.org/docs/dev/api/skimage.segmentation.html#skimage.segmentation.slic), which we import on **Line 2**. To draw the actual superpixel segmentations, scikit-image provides us with a `mark_boundaries` function which we import on **Line 3**.

From there, we import a utility function, `img_as_float` on **Line 4**, which as the name suggests, converts an image from an unsigned 8-bit integer, to a floating point data with, with all pixel values called to the range $[0, 1]$.

 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Lines 10-12 handle parsing our command line arguments. We need only a single switch, `--image` , which is the path to where our image resides on disk.

We then load this image and convert it from an unsigned 8-bit integer to a floating point data type on **Line 15**.

Now the interesting stuff happens.

We start looping over our number of superpixel segments on **Line 18**. In this case, we'll be examining three increasing sizes of segments: 100, 200, and 300, respectively.

We perform the SLIC superpixel segmentation on **Line 21**. The `slic` function takes only a single *required* parameter, which is the image we want to perform superpixel segmentation on.

However, the `slic` function also provides many optional parameters, which I'll only cover a sample of here.

The first is the `n_segments` argument which defines how many superpixel segments we want to generate. This value defaults to 100 segments.

We then supply `sigma` , which is the smoothing Gaussian kernel applied prior to segmentation.


Other optional parameters can be utilized in the function, such as `max_iter` , which the maximum number of iterations for k-means, `compactness` , which balances the color-space proximity with image space-proximity, and `convert2lab` which determines whether the input image should be converted to the L*a*b* color space prior to forming superpixels (in nearly *all* cases, having `convert2lab` set to `True` is a good idea).

Now that we have our segments, we display them using matplotlib in **Lines 24-27**.

In order to draw the segmentations, we make use of the `mark_boundaries` function which simply takes our original image and overlays our superpixel segments.

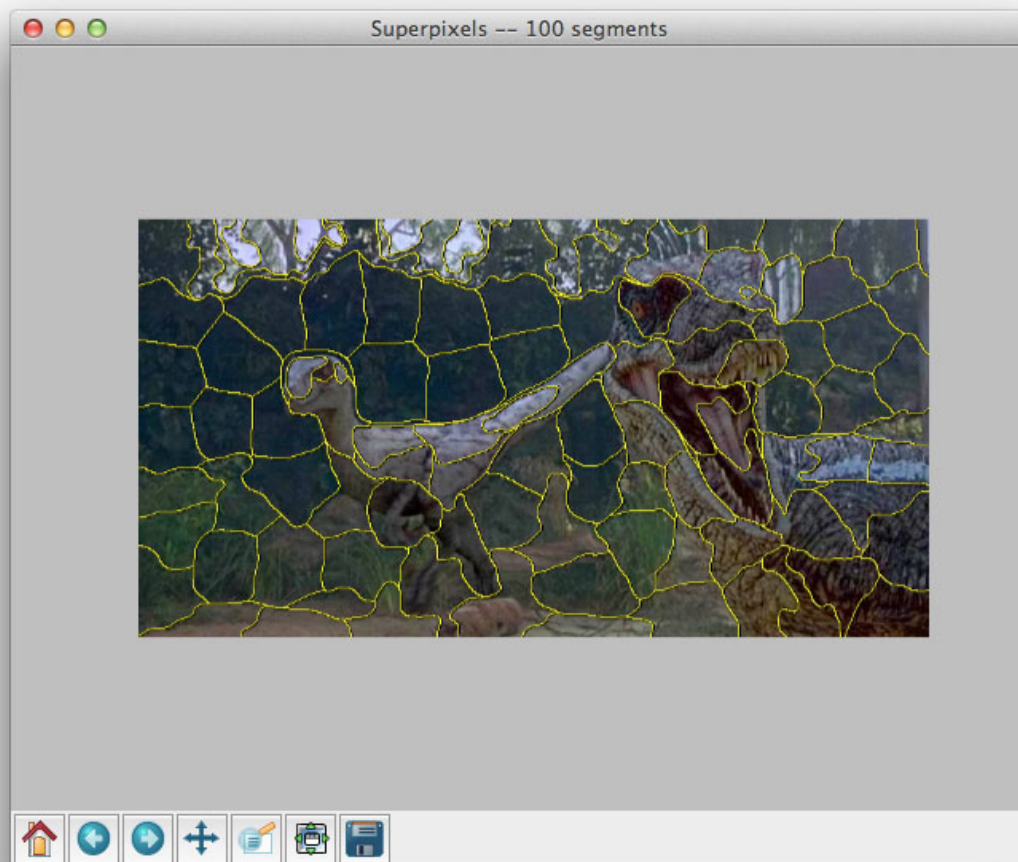
Finally, our results are displayed on **Line 30**.

Now that our code is done, let's see what our results look like.

 **Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.**
Tap to answer


```
1. | $ python superpixel.py --image raptors.png
```

If all goes well, you should see the following image:



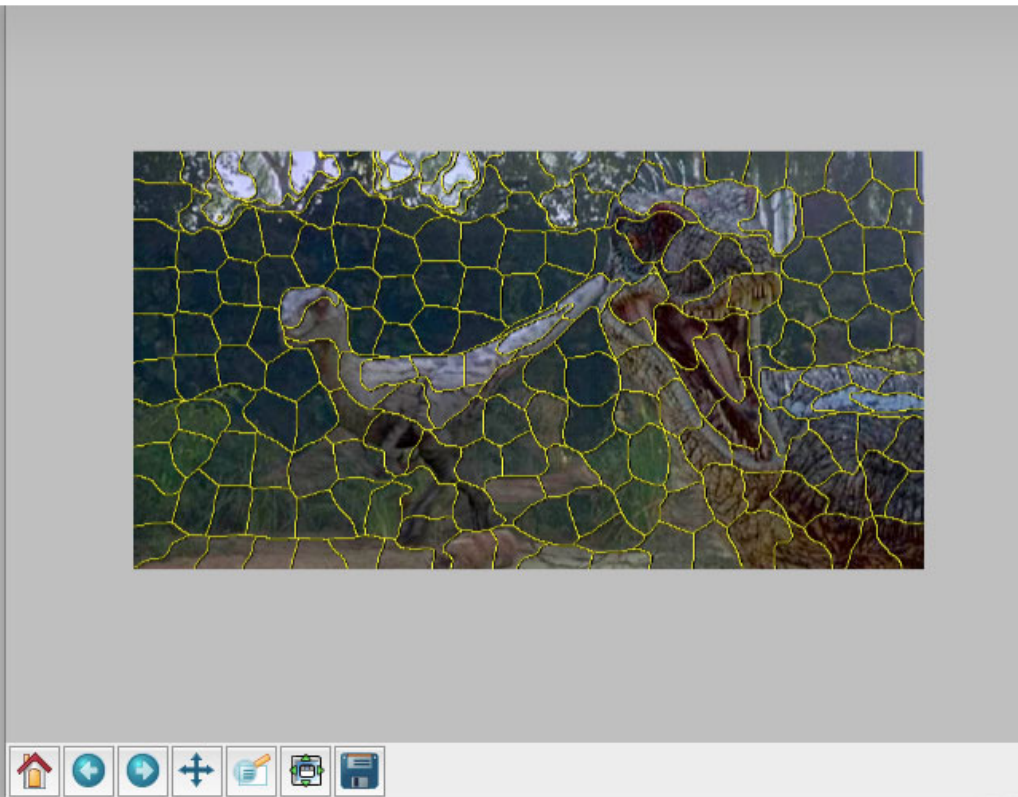
https://pyimagesearch.com/wp-content/uploads/2014/07/slic_100_segments.jpg

Figure 2: Applying SLIC superpixel segmentation to generate 100 superpixels using Python.

In this image, we have found (approximately) 100 superpixel segmentations. Notice how locally similar regions of the image, such as the scales of the Velociraptor and the shrubbery are grouped in similar superpixels.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer



https://pyimagesearch.com/wp-content/uploads/2014/07/slic_200_segments.jpg

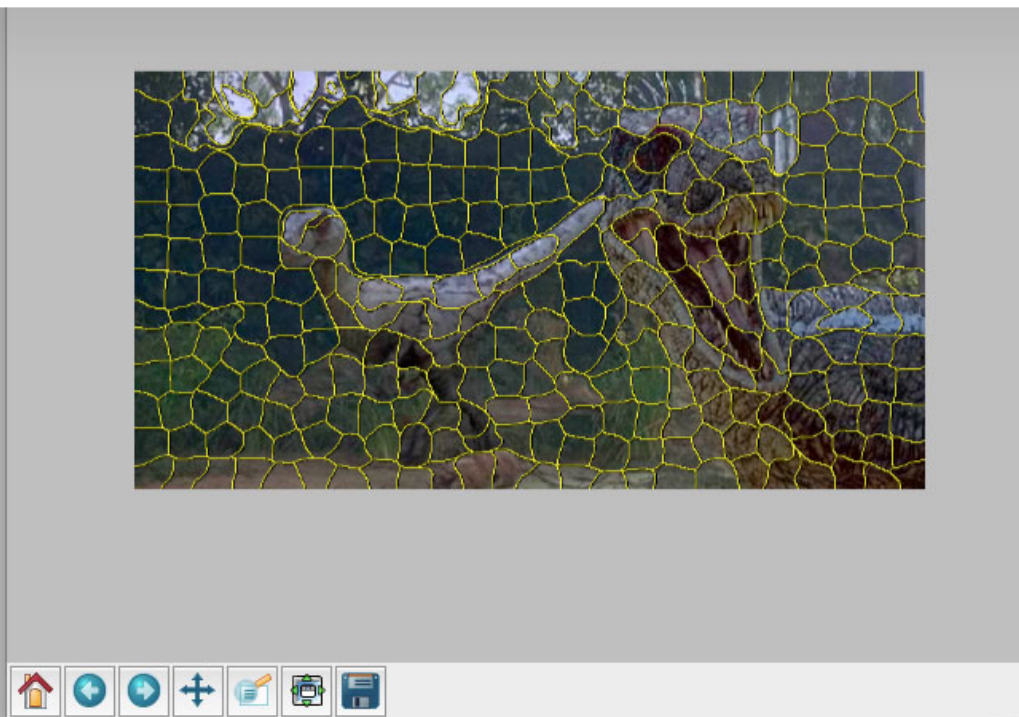
Figure 3: Applying SLIC superpixel segmentation to generate 200 superpixels using Python.

Same story here — local regions with similar color and texture distributions are part of the same superpixel group.

Finally, let's generate a very dramatic oversegmentation of the image using 300 super pixels:

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer



https://pyimagesearch.com/wp-content/uploads/2014/07/slic_300_segments.jpg

Figure 4: Applying SLIC superpixel segmentation to generate 300 superpixels using Python.

Notice how as the number of segments increases, the segments also become more rectangular and grid like. This is not a coincidence, and it can be further controlled by the optional `compactness` parameter of `slic`.

Summary

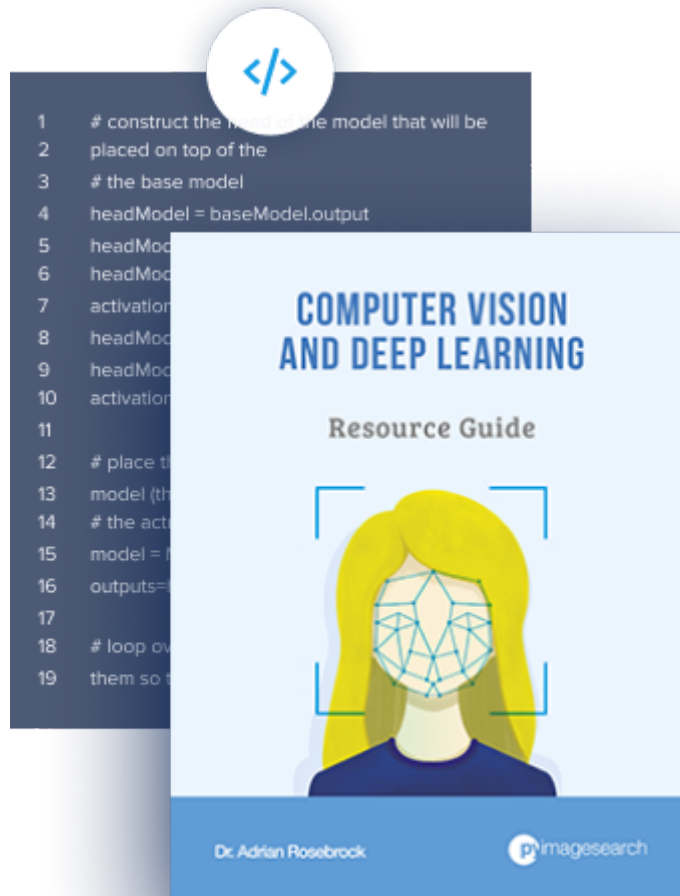
In this blog post I explained what superpixel segmentation is and how it has many benefits in the computer vision world.

Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Finally, I showed you how to utilize the Simple Linear Iterative Clustering (SLIC) algorithm to apply superpixel segmentation to your own images.

This certainly won't be the last time we discuss superpixels on this blog, so look forward to more posts in the future!



Download the Source Code and FREE 17-page Resource Guide

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer



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:

Detecting Circles in Images using OpenCV and Hough Circles

(<https://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/>)

Next Article:

OpenCV and Python Color Detection

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself. ✕

Tap to answer

77 responses to: Segmentation: A SLIC Superpixel Tutorial using Python



Wajihullah Baig

September 10, 2014 at 3:15 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-49548>)

Neatly explained. Wonderful!



Luca

October 7, 2014 at 11:21 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-82371>)

Thanks for the great explanation!

I have a question. Is there an easy way to get the neighbouring super pixels (pixels that share a border) for a super pixel of a given ID returned by slic?



Adrian Rosebrock

October 7, 2014 at 12:36 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-82469>)

Yes, but not easily. Each of the superpixels is represented by a unique integer in the “mask” returned by SLIC. I’ll write a post soon on how to obtain each of the individual superpixel segmentations

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself. ✕

Tap to answer

gee!! this is really great stuff. i just found that superpixel will be available in opencv 3.0. Really looking forward to learn more about them from you.

Thanks a lot, and keep up the good work!

**Adrian Rosebrock**

November 7, 2014 at 6:37 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-157586>)

I have a new post on superpixel algorithms and accessing each individual segment almost ready to go. It should go live towards the end of November/early December.

**Chris Viehoff**

February 27, 2015 at 4:02 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-282055>)

How do you install skimage or the module skimage.segmentation.

I get the error when I run:

```
(cv)pi@raspberrypi ~/mycode $ python superpixel.py -i ../mycode/images/trex.png
```

```
Traceback (most recent call last):
```

```
File "superpixel.py", line 3, in
```

```
from skimage.segmentation import slic
```

```
ImportError: No module named skimage.segmentation
```

**Adrian Rosebrock**

February 28, 2015 at 9:06 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-282055>)

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**Ruben Garcia**

April 2, 2015 at 1:59 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-300592>)

Hello Adrian:

I've trying to use your code.

In this code:

```
ax.imshow( mark_boundaries(image, segments) )
```

I got this error:

RuntimeError: array type not supported

Related to:

`_min_or_max_filter`

I fixed it by adding this line:

```
# Cast segments to int8!!!!
```

```
segments = segments.astype(np.int8)
```

It works fine, however computation time is >5 seconds for a 500×300 image.

Is this a normal time?

Regards.

**Adrian Rosebrock**

April 2, 2015 at 2:28 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-300601>)

Hi Ruben, 5 seconds seems pretty slow for a 500×300 image, I'm not sure why that would be.



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

August 12, 2015 at 2:11 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-345122>)

Hello there!

Very helpful tutorial. I'm just running into a small issue when I copy your example...

At `ax.imshow(mark_boundaries(image, segments))`, I'm getting that the array is not broadcastable to the correct shape. The segments array appears to be a 10x4x512 array, whereas my original image is a 10x4x512x512 array. Is there something I'm missing when finding segments with `slic`?

Thanks!

UPDATE:

Erg. Never mind. Some digging found this OTHER link of yours:

<https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/> (<https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/>)

Using `cv2's imread` and then converting to float seems to work just fine. Not sure why.



Adrian Rosebrock

August 13, 2015 at 7:07 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-345437>)

Awesome, I'm glad to hear that it worked for you!



Brian

November 17, 2015 at 7:28 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-379333>)

Thanks for a great tutorial!

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**Adrian Rosebrock**

November 18, 2015 at 6:26 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-379449>)

Just **loop over each individual superpixel**

(<https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/>), construct a mask for each, then use the mask to compute the mean of the region. Keep track of the darkest and light regions, respectively.

**Ovi**

January 28, 2016 at 3:22 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-388272>)

I am getting an error at this line:

```
ap.add_argument("-i", "-image", required = True, help = "Path to the image")
```

How it should look like if I have a path to, let's say, "C:\user\image.jpg" ?

Thanks!

**Adrian Rosebrock**

January 28, 2016 at 3:42 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-388330>)

The code itself *does not* have to change. You need to open a terminal, navigate to the code, and execute the following command (like I do in the blog post):

```
$ python superpixel.py --image raptors.png
```



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**david**

January 12, 2018 at 4:40 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-446321>)

have the same problem and cant really grasp what you are explaining please

**Adrian Rosebrock**

January 12, 2018 at 5:23 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-446326>)

No worries. Be sure to read up on **command line arguments** (<https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/>). I would also suggest Googling “terminal ” and read a few tutorials to familiarize yourself with the terminal and how to execute programs via the command line.

**rabih assaf**

May 2, 2016 at 12:51 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-396461>)

Hello, Does this method works on grayscale images?
Thank you.

**Adrian Rosebrock**

May 2, 2016 at 7:47 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-396481>)

Yes, superpixels can be computed for grayscale images using SLIC.



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

thanks,

But the algo above don't work on grayscale images.

In your opinion, i must change in the slic function parameters or what?

Thank you Adrian.

**Adrian Rosebrock**

May 3, 2016 at 5:46 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-396562>)

A quick hack is to stack grayscale images into a 3D representation, like this:

```
image = np.dstack([gray, gray, gray])
```

And then passing `image` as an input to SLIC.

**assaf**

May 3, 2016 at 8:33 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-396534>)

Hello Adrian. Is there a way to get the centers of the superpixels?

**Adrian Rosebrock**

May 3, 2016 at 5:45 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-396561>)



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**Dorin-Mirel Popescu**

July 24, 2017 at 9:43 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-430655>)

The dstack hack did the trick. Thank you Adrian 😊

**Michael Smith**

May 11, 2016 at 6:41 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-397400>)

Hello,

Great tutorial. I installed and ran your code perfectly on a stock image of a Ferrari. When I switched to a different image, I get the following error:

Segmentation fault (core dumped)

I'm running on Ubuntu 14.04, python 2.7.11, ipython 3.2.0, scikit-image 0.12.3

Do you have any advice?

**Adrian Rosebrock**

May 12, 2016 at 3:37 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-397496>)

It's hard to say without seeing the exact image, but I would suggest inserting print statements in the code to debug and determine exactly where the seg-fault is caused.



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Hi Adrian,

Such a neat tutorial! quick question, is there a way to to define the minimum area of each superpixel?

Thank you!



Adrian Rosebrock

May 21, 2016 at 8:16 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-398368>)

You can define the number of superpixels that can be extracted which in turn influences the size of the superpixel (i.e., the larger the superpixels are, the smaller their area will be). But you cannot define a minimum size directly with SLIC.



Cem

May 26, 2016 at 3:42 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-398833>)

Hi,

Great post ! Can you also make a tutorial about connecting this to CRFs?



Randy

August 3, 2016 at 8:17 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-403513>)



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer



August 4, 2016 at 10:11 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-403550>)

OpenCV's SLIC implementation is still in development (hence being in the "contrib" repo). I'm also not sure if there are Python bindings for it? The scikit-image SLIC is also very easy to use and well documented.



Shubham

October 4, 2016 at 2:48 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-407458>)

Hello,

I just downloaded your code and tried to run it on anaconda 3.5. But just could not, do I have to make any changes to the code as I am trying to put the path in the "help=Path to image" line 11 of your code. But it is giving error.

Also I tried through command line of IPython, it gives syntax error with python superpixel.py – image raptors.png.

Also, can I work on a satellite image for SLIC, as it is a grayscale image, if possible. Kindly clarify how?



Adrian Rosebrock

October 4, 2016 at 6:51 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-407470>)

You do not need to put the path to the image into the help attribute. I would suggest taking a step back and **[reading up on how command line arguments work](https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/)** (<https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/>)

Before you proceed, I am quite confident this is your issue.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**Bruce**

November 21, 2016 at 1:00 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-411311>)

Awesome

**shubham**

December 10, 2016 at 2:32 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-413086>)

I am having a numpy array of a grayscale image, I just want to convert it to 3 channel image, i just tried to convert it like this:

```
m = np.dstack(gray,gray,gray)) # here m is my original image 2d array
```

now when I pass it to slic and compile :
I get error name gray is not defined .

Kindly clarify how can I convert this image to grayscale.

**Adrian Rosebrock**

December 10, 2016 at 7:08 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-413101>)

If you are getting an error related to “gray not defined” then you do not have a variable named gray. You should double check your code that you have correctly named your variable.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

It looks like you tried to do this but I see an extra closing parenthesis in your code. You might want to check that.



shubham

December 12, 2016 at 7:16 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-413276>)

Dear Adrian,

You made my day...Thanks a lot...this was a petty mistake I was doing in np.dstack. Thanks a lot.



Adrian Rosebrock

December 12, 2016 at 10:25 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-413293>)

No problem, I'm happy I could help 😊



Noah

February 9, 2017 at 7:38 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-417337>)

Hi Adrian,

I am trying to work this out in my code right now.

What should I be declaring gray as?



Adrian Rosebrock

February 10, 2017 at 2:01 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-417337>)



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

cv2.cvtColor(image, cv2.COLOR_BGR2RGB)



shubham

December 13, 2016 at 12:59 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-413389>)

Dear Adrian,

Is there any way to work on a particular superpixel i.e. I just want to retrieve all the pixel values in a particular super pixel.



Adrian Rosebrock

December 14, 2016 at 8:35 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-413503>)

Absolutely. Please see **[this post \(https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/\)](https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/)** on accessing individual superpixels.



Ajay Khetan

December 25, 2016 at 6:40 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-414239>)

i want to track object by performing segmentation using superpixel in a video. Seeing your above post, i am bit clear with segemtation part,. Can you please explain the tracking part??
Thanx in advance



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

hi Adran,

this topic is really practical and usefull. Congratulations.
I have some questions,

how can I pass the image with the drawn segmentes as an open cv object image.
how can I save to disk the image?
are the segments contours?

thank you 😊



Adrian Rosebrock

January 28, 2017 at 6:58 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-416624>)

I'm not sure what you mean by passing an image with drawn segments, but you can access each individual superpixel via [this post \(https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/\)](https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/).



Prateeth

February 20, 2017 at 6:57 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-418089>)

Hey Adrian, Will this code help in Segmenting and accessing only the nucleus part of tissue image ? I tried using active contours but applying the texture feature and checking exactly which texture dimension is best for isolating nucleus is tough. So will the superpixel help in isolating only the nucleus of the cells in tissue image ?.. if yes, please mail me some directions. Thanks Adrian.



Adrian Rosebrock

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself. ✕

Tap to answer

in determining if superpixel segmentation would help with your given project.



Ankit

February 27, 2017 at 1:26 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-418565>)

Thanks , Simply Great .. it helped me a lot



Adrian Rosebrock

February 27, 2017 at 1:35 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-418566>)

Awesome, glad to hear it Ankit!



weiyue

March 12, 2017 at 11:30 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-420035>)

hi, I am a new man to python could you please tell me how to modify in line 11?



Amir

March 24, 2017 at 10:09 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-421118>)

Awesome tutorial! Helped me a lot!

BTW, right before the code example you wrote "Open up your favorite editor, create slic.py , and let's get coding:" and later in running explanation (and in the zip file) the filename was

"superpixel.py" So you might want to change that. 😊

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**Adrian Rosebrock**

March 25, 2017 at 9:23 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-421174>)

Thank you for letting me know about the filename issue, Amir! 😊

**Sarthak**

May 2, 2017 at 9:47 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-424280>)

Thanks for the explanation. I wanted to ask how would one extract some particular segment(s)?

**Adrian Rosebrock**

May 3, 2017 at 5:47 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-424391>)

Please see **[this tutorial \(https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/\)](https://www.pyimagesearch.com/2014/12/29/accessing-individual-superpixel-segmentations-python/)** on accessing individual superpixels.

**ambika**

June 6, 2017 at 6:22 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-426812>)

should the image whose segmentation is to be done have some specific size or resolution? I am making a project on image tampering so i need to know what exact input specifications are to be kept for the same.



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Typically we only process images that have are a maximum of 600-800 pixels along their largest dimension (i.e., width or height). Often we try to resize our images as small as possible without sacrificing accuracy (in the 200-500 pixel range). The smaller the image, the less data there is to process, therefore the algorithm can run faster.



ambika

June 7, 2017 at 1:17 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-426872>)

thanks for the info! 😊



Jess

July 26, 2017 at 11:13 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-430934>)

I wanted to know if there is a way if I can implement this into a GUI application. I managed to make a button define it for segmentation but I need to be able to perform the segmentation on an already opened image in the application. There are different buttons for different operation on the images, one of them is to be superpixel segmentation.



Adrian Rosebrock

July 28, 2017 at 9:52 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-431043>)

I would suggest you take a look at [this blog post](https://www.pyimagesearch.com/2016/05/23/opencv-with-tkinter/) (<https://www.pyimagesearch.com/2016/05/23/opencv-with-tkinter/>) where I use TKinter as a GUI. Other GUI frameworks exist such as Kivy and QT.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer



August 29, 2017 at 5:50 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-433636>)

Hi Adrian, looking for ideas to use Superpixel I found my matplotlib does not work well, even a simple plot in a 5 lines with simple or randomized images, the script stuck with no errors, warnings, nothing. The process must be stopped to run another script or anything. Could you give me a hint about what can help this works right?
Regards Thank you.



Adrian Rosebrock

August 31, 2017 at 8:35 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-433748>)

If I understand your question correctly matplotlib is not displaying the output image?



ambika

October 2, 2017 at 8:51 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-436223>)

Hi Adrian!

I am trying to divide an image into 4 superpixels. The console window displays that the image has been divided into four superpixels and when i try to access them individually, I am able to access four superpixels too! But the issue is that the image on which the boundaries have been marked of superpixels (mark_boundaries function) only three lines are printed. That is, the image is shown as if it is divided into three superpixels only. One mid line and a line dividing right half into two parts is displayed but the line dividing left half into two is not marked while it is not a whole segment but two of them.



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**Adrian Rosebrock**

October 2, 2017 at 9:21 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-436226>)

Hi Ambika — is there a particular reason why you only want to generate four superpixels? Secondly, have you followed [this tutorial](https://www.pyimagesearch.com/2017/06/26/labeling-superpixel-colorfulness-opencv-python/) (<https://www.pyimagesearch.com/2017/06/26/labeling-superpixel-colorfulness-opencv-python/>) on accessing individual superpixels?

**ambika**

October 3, 2017 at 4:06 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-436330>)

Hello Adrian!

Thanks for the quick reply.

The reason why I want only four superpixels is because in my project I need to embed another image into 4 superpixels of the host image in a way that the image information does not degrade much (PSNR i mean). Thus i need exact 4 superpixels and that too in square shape (compactness is kept large).

I have referred to your tutorial addressing about how to access individual superpixels and that too is a great tutorial and I have used it extensively inside the code. But, the step of marking the boundaries of superpixels comes before accessing individual superpixels. Also I would like to prefer to use the labelled image (returned by slic) to be used for marking the boundaries as it contains the image's segmented array.

Can you please help me out with that? If that's complicated, then can you tell me any alternative through which we can mark the boundaries to display the segments formed on the whole image?



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

You don't need the labeled image returned by SLIC. You can simply extract each superpixel. The boundary of each superpixel (given by computing contours) will give you the (x, y)-coordinates. Or you could simply compute the bounding rectangle for the superpixel.



jey jey

October 13, 2017 at 7:21 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-437354>)

hello adrian,

thank you for your tutorial,

i am a total noob super beginner in python (this my first interaction with python), i am a graphic designer more familiar with java and processing.

i tried everything in my possible but i can't make it work, i'am using IDLE on mac, i think i followed every step but i am missing something.

or can you guide me to a basic tutorial just to make this script work,

sorry for my stupid question!



Adrian Rosebrock

October 13, 2017 at 8:33 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-437360>)

Did you use one of my tutorials to install OpenCV on your Mac? If so, virtual environments cannot be used in the IDLE. Please use the command line and the script will execute.



Mario

December 22, 2017 at 10:26 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-437360>)

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

I couldn't find the explanation on why do you load the image as a float?

Is there a specific reason for that?



Adrian Rosebrock

December 26, 2017 at 4:39 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-444726>)

The SLIC algorithm assumes we are using floats as we'll be scaling the data during the clustering process. Images are normally represented by 8-bit unsigned integers. Using floats fixes this.



John

December 26, 2017 at 9:17 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-444661>)

Hi

Is it expected for the output image to have the colour balance significantly shifted? My input image is rather warm, but I get a blueish output image.

Can you please link to your raptor.jpg just for testing?

Thanks



Adrian Rosebrock

December 26, 2017 at 3:45 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-444689>)



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

**Ahmed Elmanawy**

April 14, 2019 at 8:36 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-512811>)

Thank you for your description and explain but I have one question can I use this code to get superpixels in the hyperspectral image like done on Matlab by a superpixels3 script. thanks in advance

**Adrian Rosebrock**

April 18, 2019 at 7:24 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-513468>)

Sorry, I don't have any experience with hyperspectral images.

**Roma Vardiyani**

June 25, 2019 at 8:26 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-522982>)

Hello Adrian

I did not got what you taught in argument parsing. I went through your another blog on command line arguments but i think i don't need command line arguments. I am working on Spyder installed under Anaconda. I just to read multiple images from a folder on my desktop and apply superpixeling to it. Can you suggest me something how can i proceed to execute my work? The multiple images contain both RGB and Gray Scale Images.

**Adrian Rosebrock**

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

should take the time to learn command line arguments. You will encounter them at some point in career. Take the time now to invest in that knowledge. Walk before you run.



Carlos

April 8, 2020 at 1:32 pm (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-771152>)

Hi Adrian

How can I save each segment separately? I am using in the final part of the code `cv2.imwrite('hola.jpg', cv2.bitwise_and(image, image, mask = mask))` but it only saves the last segment

Apart from this, How can I save the segments for a t-shirt for example? I need to segment the clothes for a person separately



Adrian Rosebrock

April 9, 2020 at 8:57 am (<https://www.pyimagesearch.com/2014/07/28/a-slic-superpixel-tutorial-using-python/#comment-772454>)

Are you trying to save each individual superpixel to disk? If so, it looks like you're overwriting the "hola.png" file at each iteration of the loop. Give each segment a unique filename.

Before you leave a comment...



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

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 **priority support**.
- 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!

 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Similar articles

DEEP LEARNING LIBRARIES TUTORIALS

Installing Keras with TensorFlow backend

November 14, 2016

(<https://www.pyimagesearch.com/2016/11/14/installing-keras-with-tensorflow-backend/>)

DEEP LEARNING KERAS AND TENSORFLOW TUTORIALS

How to install TensorFlow 2.0 on Ubuntu

December 9, 2019

(<https://www.pyimagesearch.com/2019/12/09/how-to-install-tensorflow-2-0-on-ubuntu/>)

DEEP LEARNING KERAS AND TENSORFLOW OPENCV TUTORIALS

OPTICAL CHARACTER RECOGNITION (OCR) TUTORIALS

OpenCV Sudoku Solver and OCR

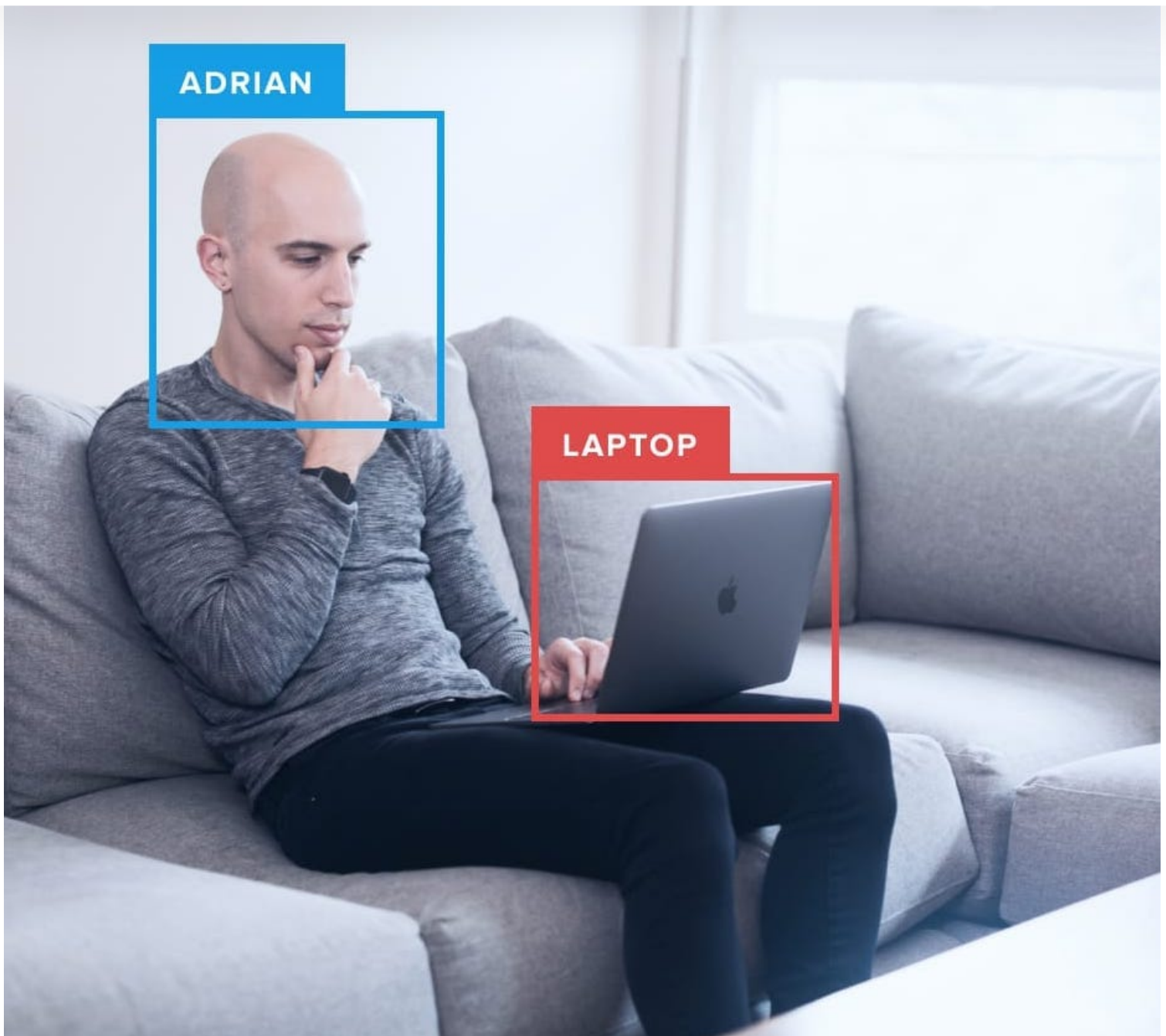
August 10, 2020

(<https://www.pyimagesearch.com/2020/08/10/opencv-sudoku-solver-and-ocr/>)



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer



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.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

Topics

[Deep Learning](https://www.pyimagesearch.com/category/deep-learning-2/)

(<https://www.pyimagesearch.com/category/deep-learning-2/>)

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

(<https://www.pyimagesearch.com/category/dlib/>)

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

(<https://www.pyimagesearch.com/category/embedded/>)

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

(<https://www.pyimagesearch.com/category/faces/>)

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

(<https://www.pyimagesearch.com/category/image-processing/>)

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

(<https://www.pyimagesearch.com/category/interviews/>)

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

(<https://www.pyimagesearch.com/category/keras/>)

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

(<https://www.pyimagesearch.com/category/machine-learning-2/>)

[Medical Computer Vision](https://www.pyimagesearch.com/category/medical/)

(<https://www.pyimagesearch.com/category/medical/>)

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

(<https://www.pyimagesearch.com/category/optical-character-recognition-ocr/>)

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

(<https://www.pyimagesearch.com/category/object-detection/>)

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

(<https://www.pyimagesearch.com/category/object-tracking/>)

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

(<https://www.pyimagesearch.com/category/opencv/>)

[Raspberry Pi](https://www.pyimagesearch.com/category/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/)

(<https://www.pyimagesearch.com/free-opencv-computer-vision-deep-learning-crash-course/>)

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

(<https://www.pyimagesearch.com/practical-python-opencv/>)

[Deep Learning for Computer Vision with Python](https://www.pyimagesearch.com/deep-learning-computer-vision-with-python/)

(<https://www.pyimagesearch.com/deep-learning-computer-vision-with-python/>)

PyImageSearch

[Get Started](https://www.pyimagesearch.com/start-here/) (<https://www.pyimagesearch.com/start-here/>)

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

(<https://www.pyimagesearch.com/opencv-tutorials-resources-guides/>)

[About](https://www.pyimagesearch.com/about/) (<https://www.pyimagesearch.com/about/>)

[FAQ](https://www.pyimagesearch.com/faqs/) (<https://www.pyimagesearch.com/faqs/>)



Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself.

Tap to answer

[gurus/](#)

[Raspberry Pi for Computer Vision](#)

(<https://www.pyimagesearch.com/raspberry-pi-for-computer-vision/>)

[Privacy Policy](#)

(<https://www.pyimagesearch.com/privacy-policy/>)

[f](https://www.facebook.com/pyimagesearch) (<https://www.facebook.com/pyimagesearch>) [t](https://twitter.com/PyImageSearch) (<https://twitter.com/PyImageSearch>)

[in](http://www.linkedin.com/pub/adrian-rosebrock/2a/873/59b) (<http://www.linkedin.com/pub/adrian-rosebrock/2a/873/59b>) [y](https://www.youtube.com/channel/UCoQK7OVcIVy-nV4m-SMck_Q/videos)

(https://www.youtube.com/channel/UCoQK7OVcIVy-nV4m-SMck_Q/videos)

© 2020 [PyImageSearch](https://www.pyimagesearch.com) (<https://www.pyimagesearch.com>). All Rights Reserved.

👋 Hey there, do you need help learning Computer Vision, Deep Learning, and OpenCV? Start by telling me about yourself. ✕

Tap to answer