

PyImageSearch Gurus Course

[🏠 \(HTTPS://GURUS.PYIMAGESEARCH.COM\)](https://gurus.pyimagesearch.com/) >

1.4.5: Cropping

Topic Progress: (<https://gurus.pyimagesearch.com/topic/translation/>)

(<https://gurus.pyimagesearch.com/topic/rotation/>) (<https://gurus.pyimagesearch.com/topic/resizing/>)

(<https://gurus.pyimagesearch.com/topic/flipping/>) (<https://gurus.pyimagesearch.com/topic/cropping/>)

(<https://gurus.pyimagesearch.com/topic/image-arithmetic/>)

(<https://gurus.pyimagesearch.com/topic/bitwise-operations/>)

(<https://gurus.pyimagesearch.com/topic/masking/>) (<https://gurus.pyimagesearch.com/topic/splitting-and-merging-channels/>)

[← Back to Lesson \(https://gurus.pyimagesearch.com/lessons/basic-image-processing/\)](https://gurus.pyimagesearch.com/lessons/basic-image-processing/)

As the name suggests, *cropping* is the act of *selecting* and *extracting* the Region of Interest (or simply, ROI), which is the part of the image we are interested in.

In a face detection application, we would want to crop the *face* from an image. And if we were developing a Python script to recognize dogs in images, we may want to *crop* the dog from the image once we have found it.

We have already utilized cropping **Module 1.2** (<https://gurus.pyimagesearch.com/lessons/image-basics/>) when we extracted the four corners of an image, but we'll review it again here for completeness.

Objectives:

Our primary objective is to become very familiar and comfortable using NumPy array slicing to crop regions from an image.

Cropping

When we crop an image, we want to remove the outer parts of the image that we are not interested in. This is commonly called selecting our *Region of Interest*, or more simply, our ROI.

We can accomplish image cropping by using NumPy array slicing (which we already introduced in **Module 1.2** (<https://gurus.pyimagesearch.com/lessons/image-basics/>)).

However, let's review it again and make sure we understand what is going on. Open up a new file, name it `crop.py`, and insert the following code:

crop.py	Python
<pre>1 # import the necessary packages 2 import cv2 3 4 # load the image and show it 5 image = cv2.imread("florida_trip.png") 6 cv2.imshow("Original", image) 7 8 # cropping an image is accomplished using simple NumPy array slices -- 9 # let's crop the face from the image 10 face = image[85:250, 85:220] 11 cv2.imshow("Face", face) 12 cv2.waitKey(0) 13 14 # ...and now let's crop the entire body 15 body = image[90:450, 0:290] 16 cv2.imshow("Body", body) 17 cv2.waitKey(0)</pre>	

Feedback

To demonstrate cropping in Python and OpenCV, we'll be using an image of myself on a trip to Florida, which we load from disk on **Lines 5 and 6**.



(https://gurus.pyimagesearch.com/wp-content/uploads/2015/03/florida_trip.jpg)

FIGURE 1: THE ORIGINAL IMAGE THAT WE WILL BE CROPPING.

Our goal here is to extract my *face* and *body* from the region using only simple cropping methods.

Under normal circumstances we would normally apply some sort of machine learning and computer vision techniques to detect my face and body in the image — however, since we are still fairly early on in the course, we are going to use our *a priori* knowledge of the image and manually supply the NumPy array slices of where the body and face reside. In future modules of this course, we'll be utilizing methods to **automatically** detect and extract faces from images. But for the time being, let's keep things simple.

We extract my face from the image on a single line of code: **Line 10**. We are supplying NumPy array slices to extract a rectangular region of the image, starting at (85, 85) and ending at (220, 250). The order in which we supply the indexes to the crop may seem counterintuitive; however, remember that OpenCV represents images as NumPy arrays with the the height first and the width second. This means that we need to supply our y-axis values before our x-axis.

In order to perform our cropping, NumPy expects four indexes:

- **Start y:** The starting y coordinate. In this case, we start at y=85.
- **End y:** The ending y coordinate. We will end our crop at y=250.
- **Start x:** The starting x coordinate of the slice. We start the crop at x=85.

- **End x:** The ending x-axis coordinate of the slice. Our slice ends at x=220.

In a similar fashion, we extract my entire *body* from the image on **Line 15** by cropping the region (0, 90) and (290, 450) from the original image.

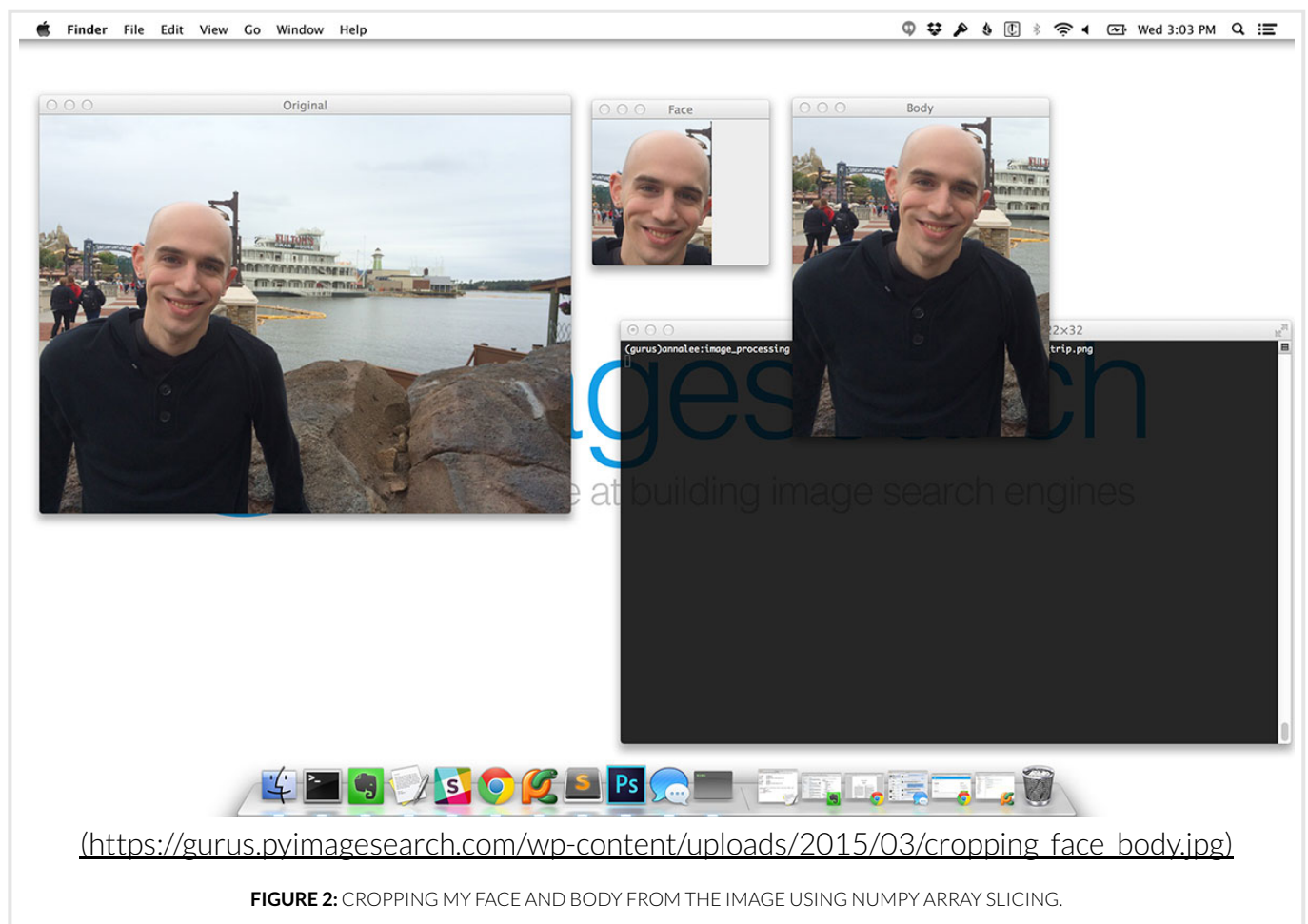
To execute the script, simply download the .zip from the bottom of this post, navigate to your source code directory, and execute the following command:

```
crop.py
```

Shell

```
1 $ python crop.py
```

After you have executed the script, you'll see the following output:



Feedback

As you can see, the face and body region of the image have been cropped!

While simple, cropping is an extremely important skill that will be utilized throughout the rest of this course. If you are still feeling uneasy with cropping, definitely take the time to practice now and hone your skills. From here on, cropping will be an *assumed skill* that you will need to understand!

Summary

In this lesson we explored how to crop regions from an image. This is a very important skill to have, so please take the time to practice with it if you are feeling uncomfortable!

Downloads:

[Download the Code](https://gurus.pyimagesearch.com/protected/code/computer_vision_basics/cro)
(https://gurus.pyimagesearch.com/protected/code/computer_vision_basics/cro)

Quizzes	Status
1	Cropping Quiz (https://gurus.pyimagesearch.com/quizzes/cropping-quiz/)

[← Previous Topic \(https://gurus.pyimagesearch.com/topic/flipping/\)](#) [Next Topic → \(https://gurus.pyimagesearch.com/topic/image-arithmetic/\)](#)

Feedback

Upgrade Your Membership

Upgrade to the *Instant Access Membership* to get **immediate access** to **every lesson** inside the PyImageSearch Gurus course for a one-time, upfront payment:

- **100%, entirely self-paced**
- Finish the course in *less than 6 months*
- Focus on the lessons that *interest you the most*
- **Access the entire course** as soon as you upgrade

This upgrade offer will expire in **30 days, 23 hours**, so don't miss out — be sure to upgrade now.

Upgrade Your Membership!
(<https://gurus.pyimagesearch.com/register/pyimagesearch-gurus-instant-access-membership-fcff7b5e/>)

Course Progress