



<https://gurus.pyimagesearch.com/>



PyImageSearch Gurus Course

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

1.4.4: Flipping

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/)

Similar to rotation, OpenCV also provides methods to *flip* an image across its x or y axis. Though flipping operations are used less often, they are still very valuable to learn — and for reasons that you may not think of off the top of your head.

For example, let's imagine that we are working for a tiny startup company that wants to build a machine learning classifier to detect faces in images. We would need some sort of dataset of example faces that our algorithm could use to “learn” what a face is. But unfortunately, the company has only provided us with a tiny dataset of 20 faces and we don't have the means to acquire more data.

So what do we do?

We apply flipping operations!

We can horizontally flip each face image (since a face is still a face, no matter if it is mirrored or not) and

While this example sounds silly and contrived, it's actually not. Powerful, data hungry, deep learning algorithms purposely use flipping to generate extra data during training time.

So as you can see, the image processing techniques you learn inside this module **really are** the building blocks for larger computer vision systems!

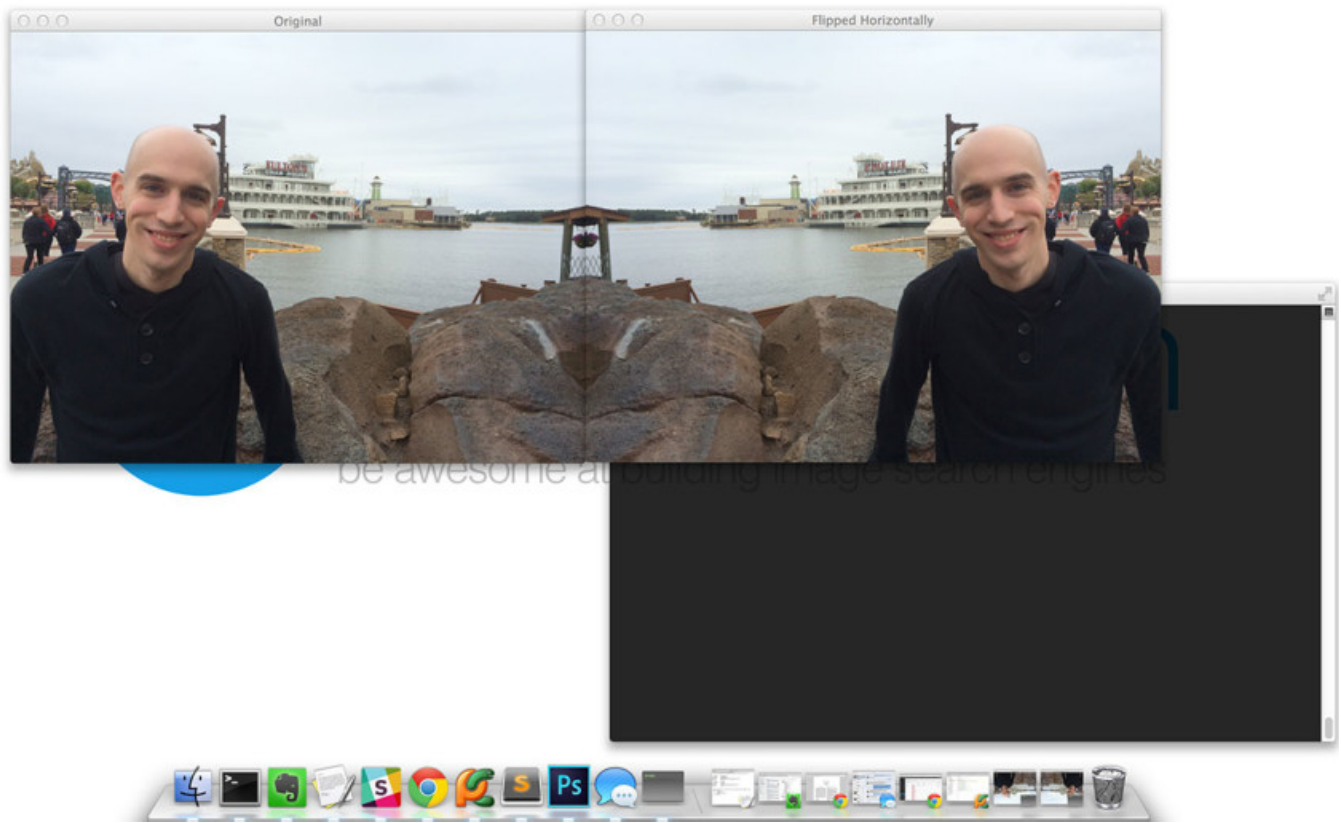
Objectives:

In this lesson you will learn how to horizontally and vertically flip an image using the `cv2.flip` function.

Flipping

Next up on our list of image transformations to explore is flipping. We can flip an image around either the x or y axis, or even both.

Flipping an image is better explained by viewing the output of an image flip, before we get into the code. Check out the image below to see an image flipped horizontally:



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

FIGURE 1: HORIZONTALLY FLIPPING AN IMAGE.

Notice how on the *left* we have our original image, and on the *right* the image has been mirrored horizontally.

Now that you see what an image flip looks like, we can explore the code:

flipping.py

Python

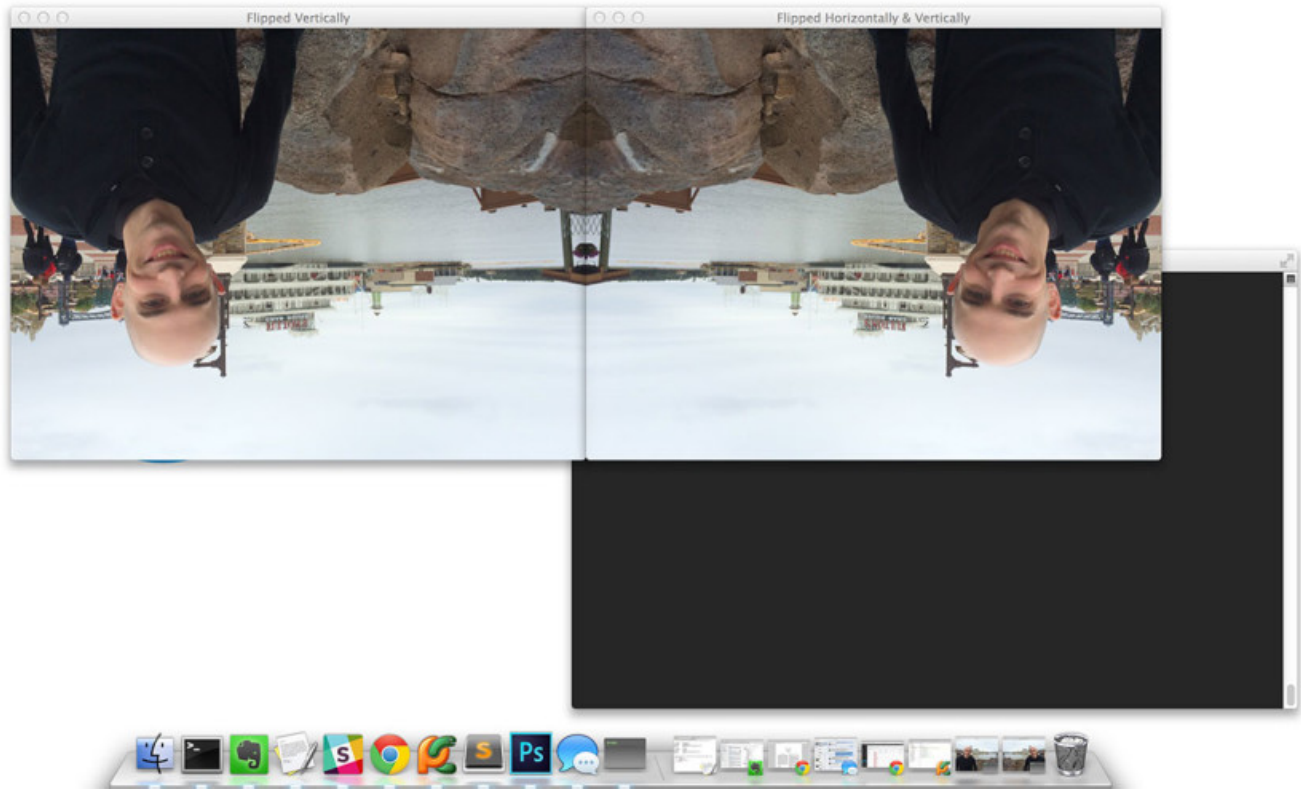
```
1 # import the necessary packages
2 import argparse
3 import cv2
4
5 # construct the argument parser and parse the arguments
6 ap = argparse.ArgumentParser()
7 ap.add_argument("-i", "--image", required=True, help = "Path to the image")
8 args = vars(ap.parse_args())
9
10 # load the image and show it
11 image = cv2.imread(args["image"])
12 cv2.imshow("Original", image)
13
14 # flip the image horizontally
15 flipped = cv2.flip(image, 1)
16 cv2.imshow("Flipped Horizontally", flipped)
17
18 # flip the image vertically
19 flipped = cv2.flip(image, 0)
20 cv2.imshow("Flipped Vertically", flipped)
21
22 # flip the image along both axes
23 flipped = cv2.flip(image, -1)
24 cv2.imshow("Flipped Horizontally & Vertically", flipped)
25 cv2.waitKey(0)
```

Lines 1-12 handle our standard procedure of importing our packages, parsing arguments, and loading our image from disk.

Flipping an image horizontally is accomplished by making a call to the `cv2.flip` function on **Line 15**. The `cv2.flip` method requires two arguments: the image we want to flip and a specific code/flag that is used to determine how we are going to flip the image.

Using a flip code value of `1` indicates that we are going to flip the image horizontally, around the y-axis (**Line 15**). Specifying a flip code of `0` indicates that we want to flip the image vertically, around the x-axis (**Line 19**). Finally, using a negative flip code (**Line 20**) flips the image around both axes.

Take look at **Figure 2** below to see our image flipped vertically, and then flipped around both axes:



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

FIGURE 3: (LEFT) OUR ORIGINAL IMAGE FLIPPED VERTICALLY. (RIGHT) THE IMAGE FLIPPED BOTH HORIZONTALLY AND VERTICALLY.

Feedback

Flipping an image is very simple — perhaps one of the most simple examples in this module!

Next up, we'll go over cropping an image and how to extract regions of an image using NumPy array slices.

Summary

Admittedly, image flipping is one of the easiest concepts we'll cover inside this course. However, just because a concept is very simple *does not* mean that it's not used for more powerful purposes. As I mentioned in the introduction to this topic, flipping is consistently used in machine learning to generate more training data samples, thus creating more powerful and robust image classifiers.

Downloads:

Download the Code (https://gurus.pyimagesearch.com/protected/code/computer_vision_basics/flipping.zip).

11/30/19, 8:47 PM

1	Flipping Quiz (https://gurus.pyimagesearch.com/quizzes/flipping-quiz/)
---	---

[← Previous Topic \(https://gurus.pyimagesearch.com/topic/resizing/\)](https://gurus.pyimagesearch.com/topic/resizing/) [Next Topic → \(https://gurus.pyimagesearch.com/topic/cropping/\)](https://gurus.pyimagesearch.com/topic/cropping/)

Course Progress

Ready to continue the course?

Click the button below to **continue your journey to computer vision guru**.

[I'm ready, let's go! \(/pyimagesearch-gurus-course/\)](/pyimagesearch-gurus-course/)

Resources & Links

- [PyImageSearch Gurus Community \(https://community.pyimagesearch.com/\)](https://community.pyimagesearch.com/)
- [PyImageSearch Virtual Machine \(https://gurus.pyimagesearch.com/pyimagesearch-virtual-machine/\)](https://gurus.pyimagesearch.com/pyimagesearch-virtual-machine/)
- [Setting up your own Python + OpenCV environment \(https://gurus.pyimagesearch.com/setting-up-your-python-opencv-development-environment/\)](https://gurus.pyimagesearch.com/setting-up-your-python-opencv-development-environment/)
- [Course Syllabus & Content Release Schedule \(https://gurus.pyimagesearch.com/course-syllabus-content-release-schedule/\)](https://gurus.pyimagesearch.com/course-syllabus-content-release-schedule/)
- [Member Perks & Discounts \(https://gurus.pyimagesearch.com/pyimagesearch-gurus-discounts-perks/\)](https://gurus.pyimagesearch.com/pyimagesearch-gurus-discounts-perks/)
- [Your Achievements \(https://gurus.pyimagesearch.com/achievements/\)](https://gurus.pyimagesearch.com/achievements/)
- [Official OpenCV documentation \(http://docs.opencv.org/index.html\)](http://docs.opencv.org/index.html)

Your Account

- [Account Info \(https://gurus.pyimagesearch.com/account/\)](https://gurus.pyimagesearch.com/account/)
- [Support \(https://gurus.pyimagesearch.com/contact/\)](https://gurus.pyimagesearch.com/contact/)
- [Logout \(https://gurus.pyimagesearch.com/wp-login.php?action=logout&redirect_to=https%3A%2F%2Fgurus.pyimagesearch.com%2F&_wpnonce=5736b21cae\)](https://gurus.pyimagesearch.com/wp-login.php?action=logout&redirect_to=https%3A%2F%2Fgurus.pyimagesearch.com%2F&_wpnonce=5736b21cae)

Q Search

© 2018 PyImageSearch. All Rights Reserved.

Feedback