



# Compare Two Images and Highlight Differences using Python

Vyom Srivastava | June 21, 2021 |



[Home](#) » Compare Two Images and Highlight Differences using Python

**OpenCV** is a powerful and highly optimized open-source library developed and released under the BSD 3-clause license. It's available for cross-platform (Linux, macOS, ios, windows, and android) and supports multiple languages (C++, Java, and python ) it's very versatile. It is one of the best methods to compare two images and highlight differences using Python.

It was developed for machine learning, image processing, 3D reconstruction, object detection, and many more but nowadays it plays a major role in real-time application which is very important in this era, with its real-time operations one can process images and videos, identify objects and even it can tell the difference in handwriting or two very much similar images. OpenCV applications are only limited by our imagination.

In this article, we are going to use OpenCV package and use it to compare two images and highlight differences using Python.

For example, let's take these two webpage screenshots:



Data Science, ML &amp; AI, Programming

ML &amp; AI, Programming

Data Science, Programming



April 22, 2021

## How to use AWS Boto in Python (AWS S3 SDK)?

AWS S3 is one of the most popular services provided by AWS. The S3 stands for Simple Storage Service and lets you fetch or upload...

Automation, Programming

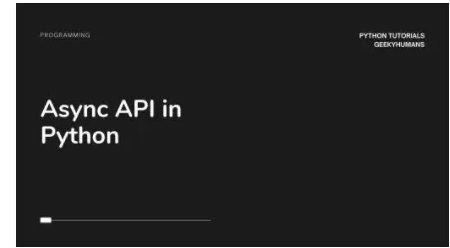


April 18, 2021

## Python tools for full stack automation testing

Learn about the best Python libraries for full stack automation testing on the basis of their features and flexibility. Python is an open-source, high-level, and...

Automation



January 26, 2021

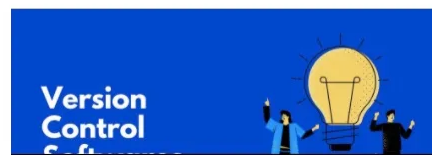
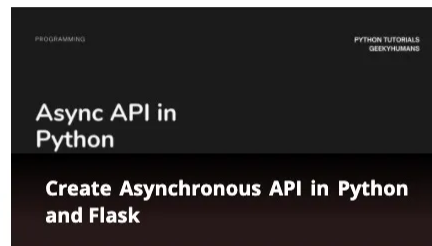
## Create Asynchronous API in Python and Flask

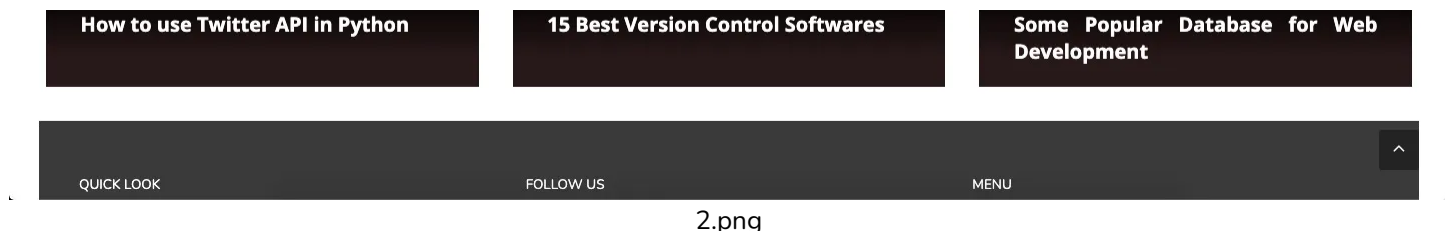
In this tutorial, we are going to create asynchronous API in Python using Flask and Asyncio. After that, we'll be testing the API as well....

Programming



1.png





2.png

In the first image, we have 3 articles and in the second one, we have 6 articles. We are going to highlight and find the difference between these two screenshots using OpenCV. Now we are going to assume we have two of these webpage screenshots somewhere in our system memory.

There are 2 fundamental elements we have to keep in mind:

1. Both of the images have has to be the same size and channels.



2. Each of the pixels has to be the same value.

We can do it in the following seven steps:

- Load the original image and the second one
- Check the size of the images
- Find what's different between two images
- Convert them into grayscale
- Increasing the size of the differences
- Finding the counters for the changes
- Displaying box around the changes

### Contents [hide]

#### 1 Procedure



- 2 Prerequisites
- 3 Step – 1: Installing the Package
- 4 Step 2: Importing the Dependencies
- 5 Step 3: Convert the Image into Grayscale
- 6 Output
- 7 Final Words

## Procedure

First, we are going to load the images in the system or software that we are going to use to use in this operation you can use what you prefer there are a lot of IDE on the web like PyCharm, Microsoft visual studio code, Eclipse, etc. they are all good you can use any one of them. You can use another programming language to perform this but in this tutorial, we are going to use python it is a very friendly and beginner-level language most people start learning programming with this language and it has lots of libraries that we can use.

Now let's start the process:



Let's start by checking if they have the same size and channels. if both of the images have the same size and channels, we have to continue forward with the operation, if both of the images don't have the same size and channels then they are not equal.

If both of the images have the same sizes and channels, we can proceed by subtracting them. The operation `cv2.subtract(image1,image2)` instead of arithmetic operation and simply subtract each pixel from the first image to the value of the corresponding pixel in the second image.

Now let's start coding:

## Prerequisites

- OpenCV Package



- Knowledge of Python

## Step – 1: Installing the Package

Now open the terminal and install the required packages using the below commands:

```
python3 -m pip install imutils  
python3 -m pip install opencv-python
```

## Step 2: Importing the Dependencies

Now open your favourite code editor and paste the below code:

```
import cv2  
import imutils
```

Then we need to import the required libraries including openCV as well as utils. We can easily install `imutils` using the “`pip install imutils`” command.

```
#get the images you want to compare.  
original = cv2.imread("1.png")  
new = cv2.imread("2.png")  
#resize the images to make them small in size. A bigger size image may take a significant time  
#more computing power and time
```



```
original = imutils.resize(original, height = 600)
new = imutils.resize(new, height = 600)
```

After that we need to read the images from the system memory, we need to make sure that both the images are placed in the same folder as the code that we are writing, else you need to provide the location path of the images. Once the images are fully loaded, we need to resize the images to a more manageable size (high = 960 pixels in this case). Now we can also work with the original images themselves, although sometimes images are really large in size for that reason it may take significantly more time to process the images.

```
#create a copy of original image so that we can store the
#difference of 2 images in the same on
diff = original.copy()
cv2.absdiff(original, new, diff)
```

## Step 3: Convert the Image into Grayscale

Now we have to use the function `absdiff` that helps to find the absolute difference between the pixels of the two image arrays. With the help of this function, we will be able to calculate per-element exact difference between two arrays. The difference is returned in the third argument.

```
#converting the difference into grayscale images
gray = cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)

#increasing the size of differences after that we can capture them all
for i in range(0, 3):
    dilated = cv2.dilate(gray.copy(), None, iterations=i+1)
```

In this section of the code, we first convert the images into grayscale that help in making and applying different types of operations on the images really easy and then we apply morphological operation (Morphological Operations is a broad set of image processing

operations that can process digital images based on their shapes and sizes.) on the images to merge the pixel together, as we are not really interested in the exact difference but we are really interested in the region of difference on images.

```
#threshold the gray image to binary it. Anything pixel that has
#value higher than 3 we are converting to white
#(remember 0 is black and 255 is exact white)
#the image is called binarised as any value lower than 3 will be 0 and
# all of the values equal to and higher than 3 will be 255
(T, thresh) = cv2.threshold(dilated, 3, 255, cv2.THRESH_BINARY)

# now we have to find contours in the binarized image
cnts = cv2.findContours(thresh, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
```

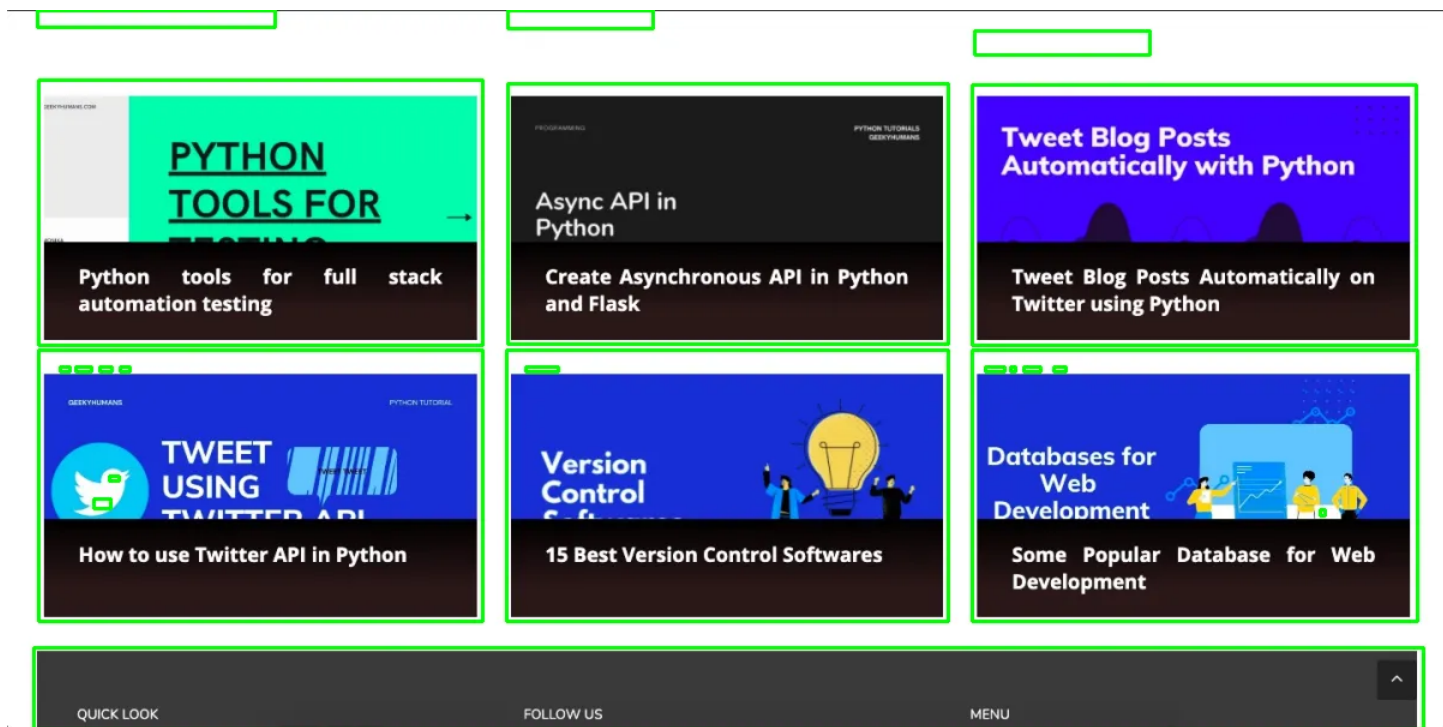
Online five we threshold the dilated grayscale image to binarize it. Any pixel that has a value higher than three we are converting into white (we have to remember pixel value 0 is black and 255 is exact white) and the image is called binarized as any value lower than 3 will be 0 and all values similar to and higher than 3 will be converted to 255. After that, we can use the OpenCV function to find Contours on the binarized image.

```
for c in cnts:
    # nicely fitting a bounding box to the contour
    (x, y, w, h) = cv2.boundingRect(c)
    cv2.rectangle(new, (x, y), (x + w, y + h), (0, 255, 0), 2)

#remove comments from below 2 lines if you want to
#for viewing the image press any key to continue
#simply write the identified changes to the disk
cv2.imwrite("changes.png", new)
```

## Output





In the first of the syntax, we loop through all the contours we have found in the image and we can try to find the rectangular green color bounding boxes for the same images. On the third line of the syntax, we have to make green color rectangles on the green bounding boxes and finally, we can save the image on the disk on the last line of the syntax. These green rectangle boxes will help you to compare two images and highlight differences using Python. Easy right?

CLONE FROM GITHUB

## Final Words

So, this is how you can compare two images and highlight differences using Python. Let us know if you guys have any questions/comments regarding this. The comparison is just the first step in the world of image processing, you can achieve a lot more with OpenCV.

Here are some useful tutorials that you can read:

- [How to convert an image to 8-bit image](#)
- [Create various charts using Python](#)
- [Password generator using Python and Secrets](#)
- [Create Asynchronous API using Flask](#)



- [Tweet blog posts automatically using Python and Twitter API](#)

Spread the love



[Previous](#)

[Draw Various Types of Charts and Graphs using Python](#)

[How to Secure Files using Python?](#)

[Next](#)

## Leave a Comment

☐ Save my name, email, and website in this browser for the next time I comment.



POST COMMENT

Holistic Detection OpenCV Python With...

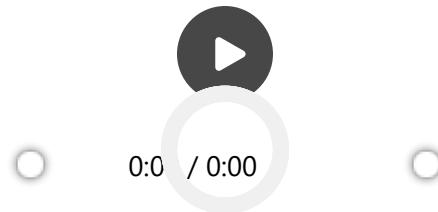




report this ad



report this ad



Holistic Detection OpenCV Python With...



## RECENT POSTS

How to Publish Python Package on PyPI

Classes and Objects in Python

Test Your Typing Speed Using Python

How to Update your Mac Address using Python

Download Youtube Videos using Python

Schedule Python Scripts with Apache Airflow

How to create a Word Guessing Game in Python

How to Deploy Flask API on Heroku?



report this ad



















## QUICK LOOK

[Home Page](#)

[Contact Us](#)

[Privacy Policy](#)

[Write For Us](#)

[About](#)

## FOLLOW US

## RECENT POSTS

[How to Publish Python Package on PyPI](#)

[Classes and Objects in Python](#)

[Test Your Typing Speed Using Python](#)



## How to Update your Mac Address using Python

### MENU

Programming

Web Development

App Development

FOSS

Automation

Dev Tools

Data Science

ML & AI

OS

Blockchain

Python Jobs

© 2022 GeekyHumans | All Rights Reserved | Write For Us

