Python | Grayscaling of Images using OpenCV

GeeksforGeeks

Data Structures Algorithms Interview Preparation Topic-wise Practice C++ Java Python

Grayscaling is the process of converting an image from other color spaces e.g. RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white.

Importance of grayscaling

- **Dimension reduction:** For example, In RGB images there are three color channels and has three dimensions while grayscale images are single-dimensional.
- **Reduces model complexity:** Consider training neural article on RGB images of 10x10x3 pixel. The input layer will have 300 input nodes. On the other hand, the same neural network will need only 100 input nodes for grayscale images.
- **For other algorithms to work:** Many algorithms are customized to work only on grayscale images e.g. Canny edge detection function pre-implemented in OpenCV library works on Grayscale images only.

Let's learn the different image processing methods to convert a colored image into a grayscale image.

Attention geek! Strengthen your foundations with the **Python Programming Foundation** Course and learn the basics.

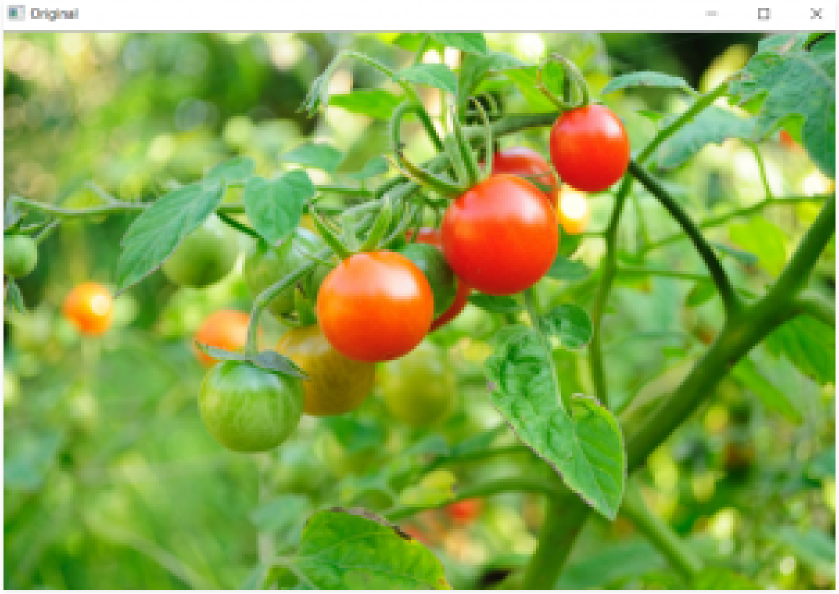
To begin with, your interview preparations Enhance your Data Structures concepts with the **Python DS** Course. And to begin with your Machine Learning Journey, join the **Machine Learning - Basic Level Course**

Method 1: Using the cv2.cvtColor() function

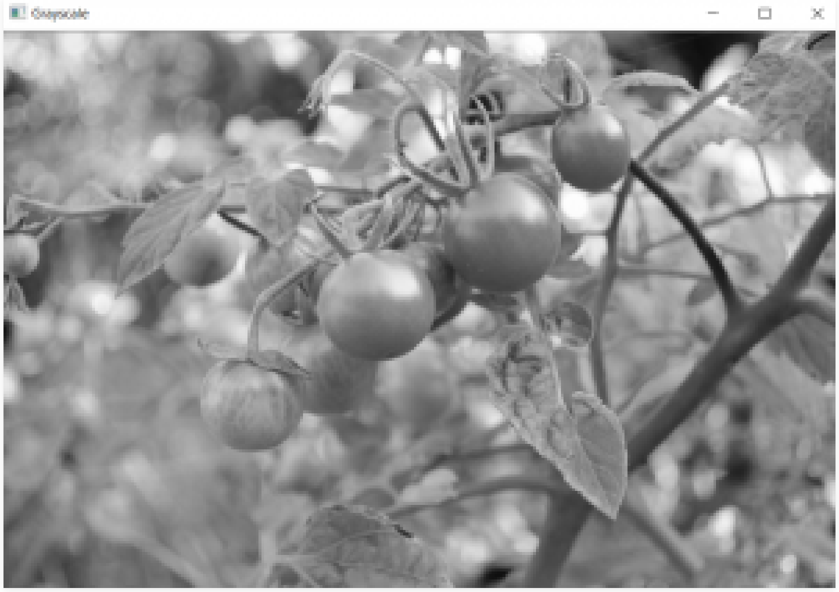
Python3

```
# import opencv
import cv2

# Load the input image
image = cv2.imread('C:\\Documents\\full_path\\tomatoes.jpg')
cv2.imshow('Original', image)
cv2.waitKey(0)

# Use the cvtColor() function to grayscale the image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2.imshow('Grayscale', gray_image)
cv2.waitKey(0)

# Window shown waits for any key pressing event
cv2.destroyAllWindows()
```

Input image:



Output Image:

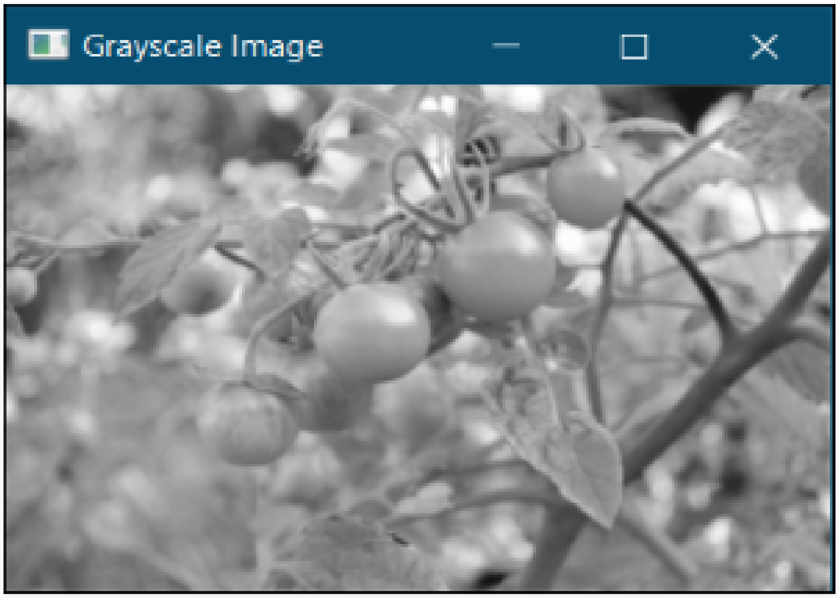


Method 2: Using the cv2.imread() function with flag = zero

Python3

```
# Import opencv
import cv2

# Use the second argument or (flag value) zero
# that specifies the image is to be read in grayscale mode
img = cv2.imread('C:\\Documents\\full_path\\tomatoes.jpg', 0)

cv2.imshow('Grayscale Image', img)
cv2.waitKey(0)

# Window shown waits for any key pressing event
cv2.destroyAllWindows()
```

Output Image:

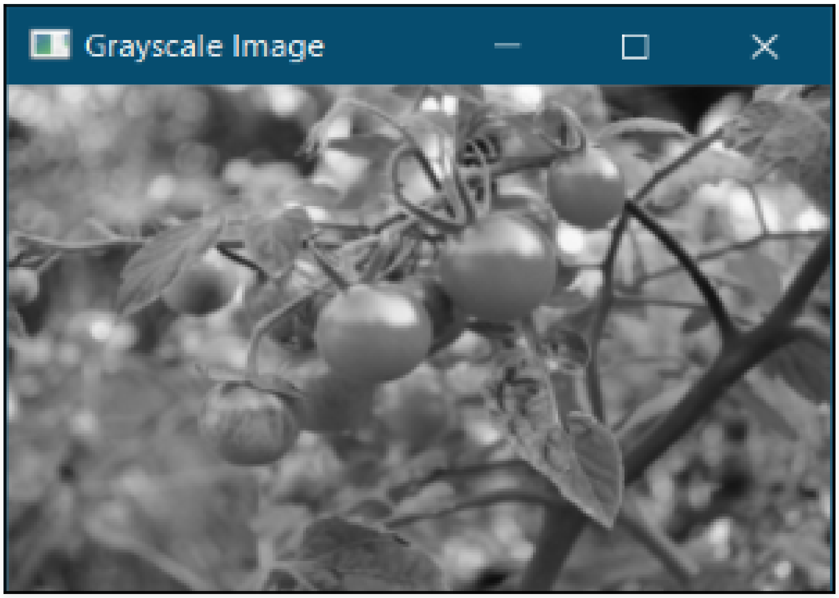


Method 3: Using the pixel manipulation (Average method)

Python3

```python
# Import opencv
import cv2

# Load the input image
img = cv2.imread('C:\\Documents\\full_path\\tomatoes.jpg')

# Obtain the dimensions of the image array
# using the shape method
(row, col) = img.shape[0:2]

# Take the average of pixel values of the BGR Channels
# to convert the colored image to grayscale image
for i in range(row):
    for j in range(col):
        # Find the average of the BGR pixel values
        img[i, j] = sum(img[i, j]) * 0.33

cv2.imshow('Grayscale Image', img)
cv2.waitKey(0)

# Window shown waits for any key pressing event
cv2.destroyAllWindows()
```

Output Image:



Hope you have understood the above discussed image processing techniques to convert a colored image into a grayscale image in Python!

Like 11

Next

Image Processing in Python (Scaling, Rotating, Shifting and Edge Detection)

RECOMMENDED ARTICLES Page : 1 2 3

01 Arithmetic Operations on Images using OpenCV | Set-2 (Bitwise Operations on Binary Images)

05 How to Detect Shapes in Images in Python using OpenCV?
13, Jan 21