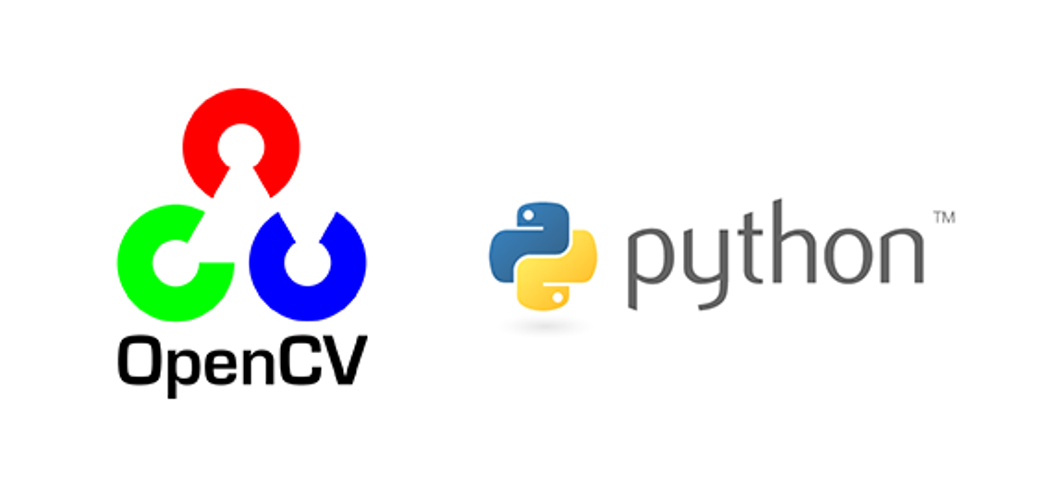
**Read, Write and Display images**

**OpenCV-Python:**

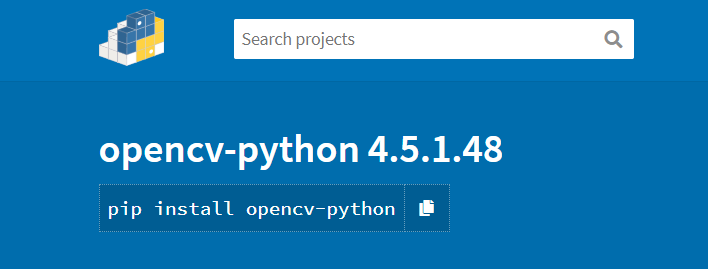
Open Source Computer Vision called as OpenCV. OpenCV-Python is a library of Python bindings designed to solve computer vision problems. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.

****

**Steps to Install OpenCV on Windows**

1. Install Python on your system
2. Install pip
3. Install OpenCV library using pip

After the installation of the Python and pip, we can directly install the OpenCV library and start using them. To install the library, we need to enter the given command in the terminal.



### **How to Verify the OpenCV Installation is Complete?**

To check if everything is installed properly, follow the below steps:

Open the command prompt in our system.

Start the Python Jupyter notebook by typing Jupyter notebook and then hit enter. You will be inside the Python Jupyter notebook where we can execute our Python code. Then we Import the cv2 package which is the name of the OpenCV module. Type “import cv2” and hit enter.

If you don’t get any error after completing these steps, then you have successfully installed the Python and OpenCV library.

**Steps to Read, Display and Save Image in OpenCV**

The main objective of this Blog is to understand how to load and display an image in OpenCV using Python.

**Goals**

Here, we will learn how to read an image, how to display it and how to save it back

we will learn these functions: **cv2.imread()**, **cv2.imshow()**, **cv2.imwrite()**

### **Reading Images**

To read the contents of an image, we have a function **cv2.imread()**. The image should be in the same directory. If not, then the full path of the image should be given.

To read an image, opencv provide a **cv2.imread()** function . This function support various file format like JPG, BMP, PNG, JPEG, TIFF etc.

The cv2.imread() function returns a *Numpy array* representing the image.

The first argument is the file name. The image should be in the working directory (or) a full path of the image should be given.

**Syntax:**

**cv2.imread(path, flag)**

**Parameters**:

**path**: The image should be in the working directory or a full path of image should be given.

**flag**: The flags option is used to control how the image is read.

**Code:**

# pip install opencv-python

#Import cv2 library after install opencv-python

import cv2

import numpy as np

import os

os.chdir('E:\\prasad\\SMIIT\\Opencv\\OpenCV-Reading-Writing-and-Displaying-Images') # for change Directory

# Read the given image

img=cv2.imread('22.jpg',cv2.IMREAD\_COLOR)

# OR

# img=cv2.imread('22.jpg',1)

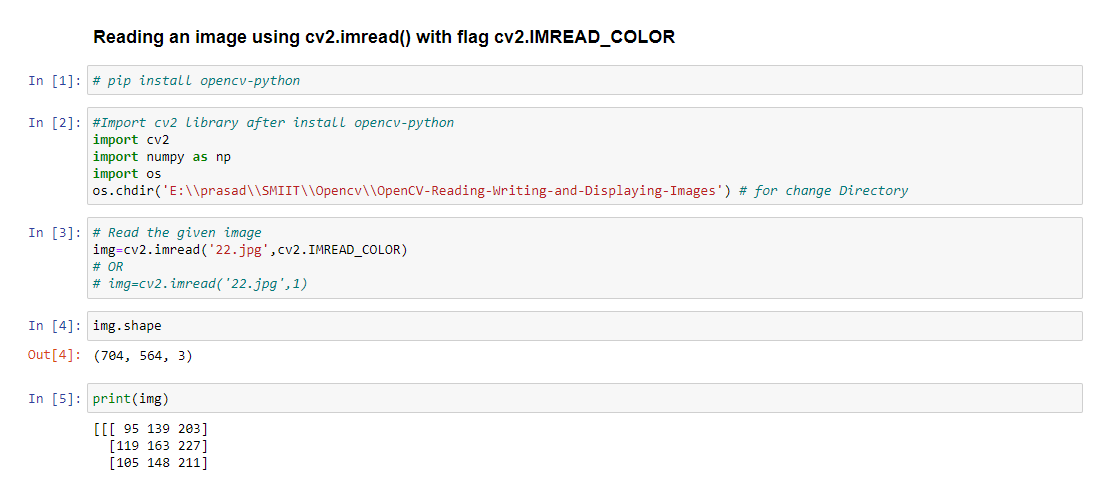
img.shape

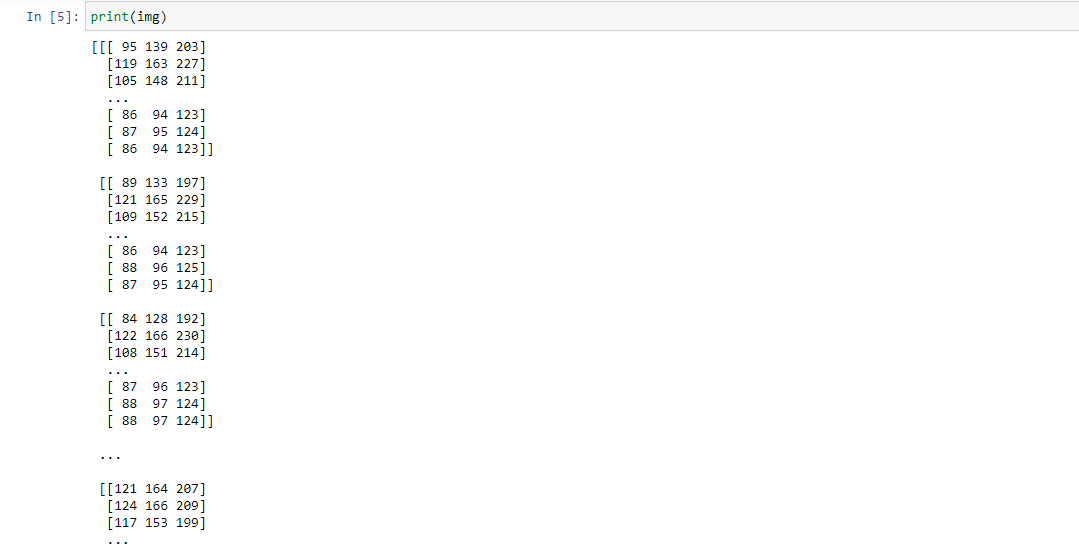
print(img)

type(img)

print(img.max())

print(img.min())

****

****

****

### **Reading an image using cv2.imread() with flag cv2.IMREAD\_GRAYSCALE**

**Code:**

# Read the given image

img=cv2.imread('22.jpg',cv2.IMREAD\_GRAYSCALE)

# OR

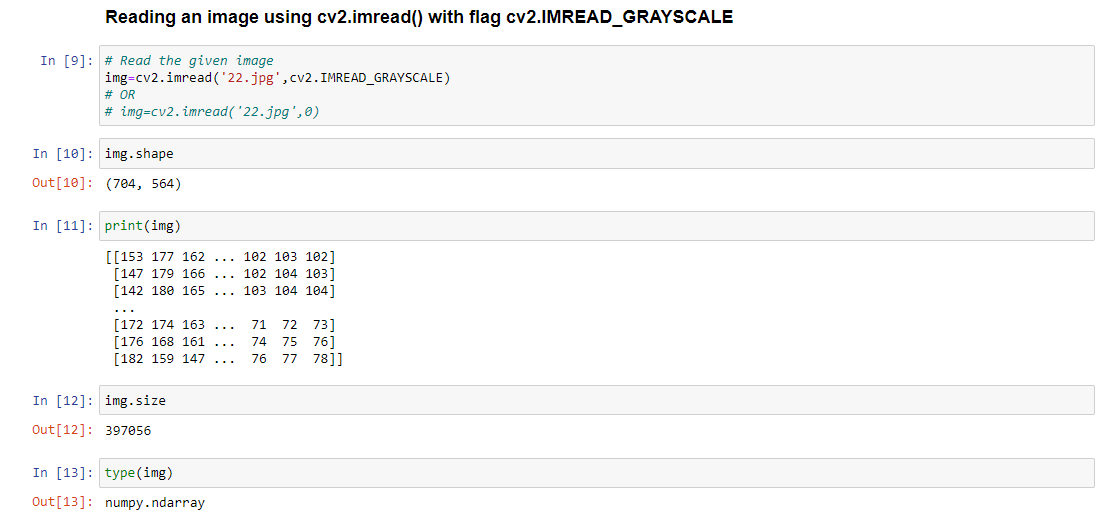
# img=cv2.imread('22.jpg',0)

img.shape

print(img)

img.size

type(img)



**Display an image:**

Use the function **cv2.imshow()** to display an image in a window.

### **Syntax**

**cv2.imshow(window\_name, image)**

**window\_name**: A string representing the name of the window in which image to be displayed.

**image**: It is the image that is to be displayed.

**Code:**

**# Display Color image using cv2.imshow()**

**cv2.imshow('Girl',img)**

**#wait to close window**

**cv2.waitKey(0)**

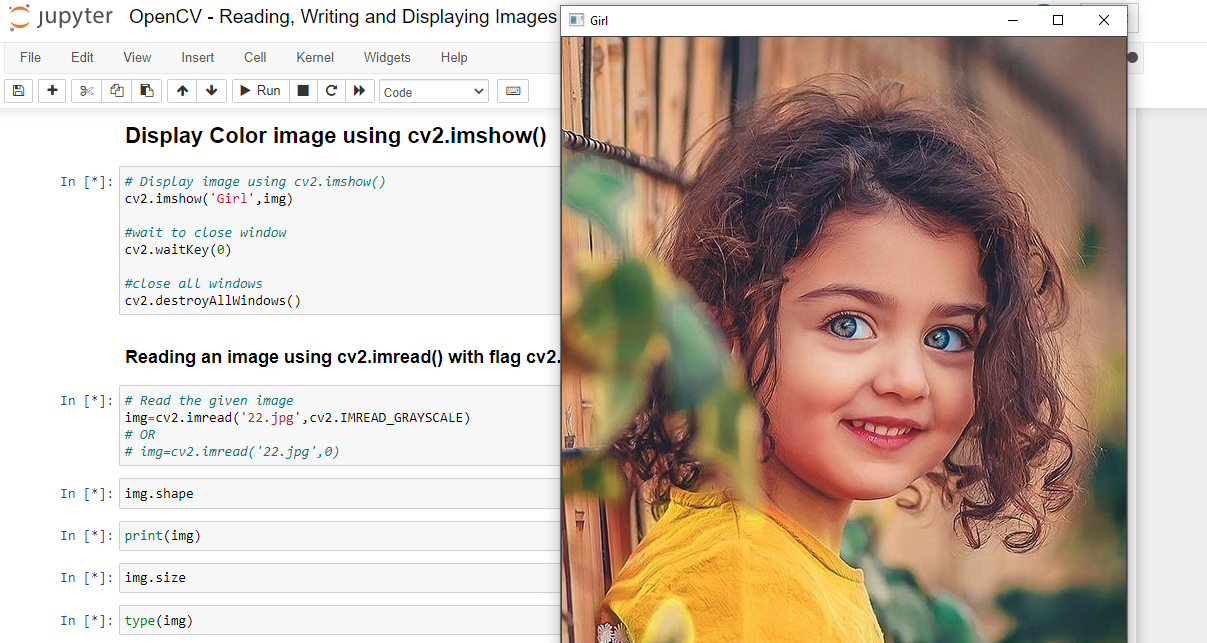
**#close all windows**

**cv2.destroyAllWindows()**

**cv2.waitKey()** is a keyboard binding function. Its argument is the time in milliseconds. The function waits for specified milliseconds for any keyboard event. If you press any key in that time, the program continues. If 0 is passed, it waits indefinitely for a keystroke.

**cv2.destroyAllWindows()** simply destroy all the windows we created. If you want to destroy any specific window, use the function cv2.destroyWindow() where you pass the exact window name as the argument.

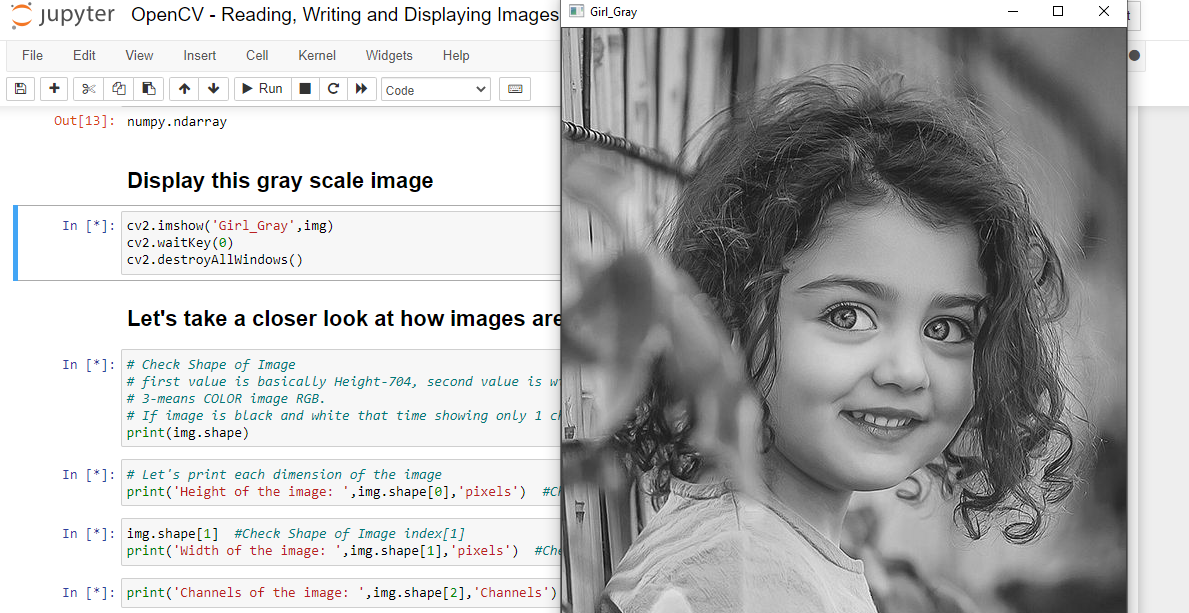
**Output:**



## **Display this gray scale image**

**Code:**

cv2.imshow('Girl\_Gray',img) cv2.waitKey(0) cv2.destroyAllWindows()



## **Let's take a closer look at how images are stored**

**Print (img.shape)**

**(704, 564, 3)**

Check Shape of Image

first value is basically Height-704, second value is width-564 and third value is channels-3.

channels-3 means COLOR image RGB.

If image is black and white that time showing only 1 channels.

**Let's print each dimension of the image**

print ('Height of the image: ',img.shape[0],'pixels')

Check Shape of Image index[0]

**Output:** Height of the image: 704 pixels

print ('Width of the image: ',img.shape[1],'pixels')

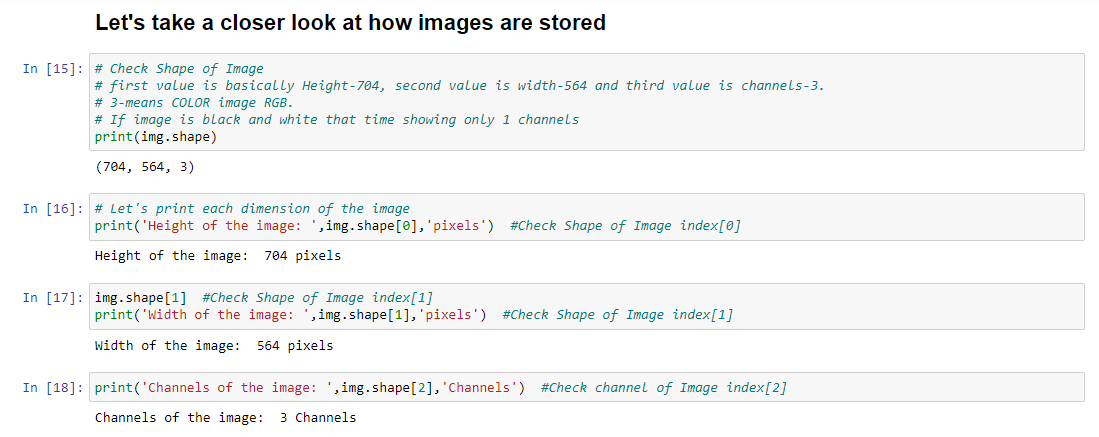
Check Shape of Image index[1]

**Output:** Height of the image: 704 pixels

print (Channels of the image: ',img.shape[2],'Channels')

Check Channels of Image index[2]

**Output:** Channels of the image: 3 Channels



## **Saving or Writing image: cv2.imwrite()**

**cv2.imwrite()** method is used to save an image to any storage device. This will save the image according to the specified format in the current working directory.

**Syntax**

**cv2.imwrite(filename, image)**

**filename**: A string representing the file name. The filename must include image format like: .jpg, .png, etc.

**image**: It is the image that is to be saved.

**Code:**

cv2.imwrite('output.jpg',img) # Save image in current location

## **Resize Image**

# Read the given image

img=cv2.imread('27.jpg')

# Resize Image

img=cv2.resize(img,(1080,720))

#show Image

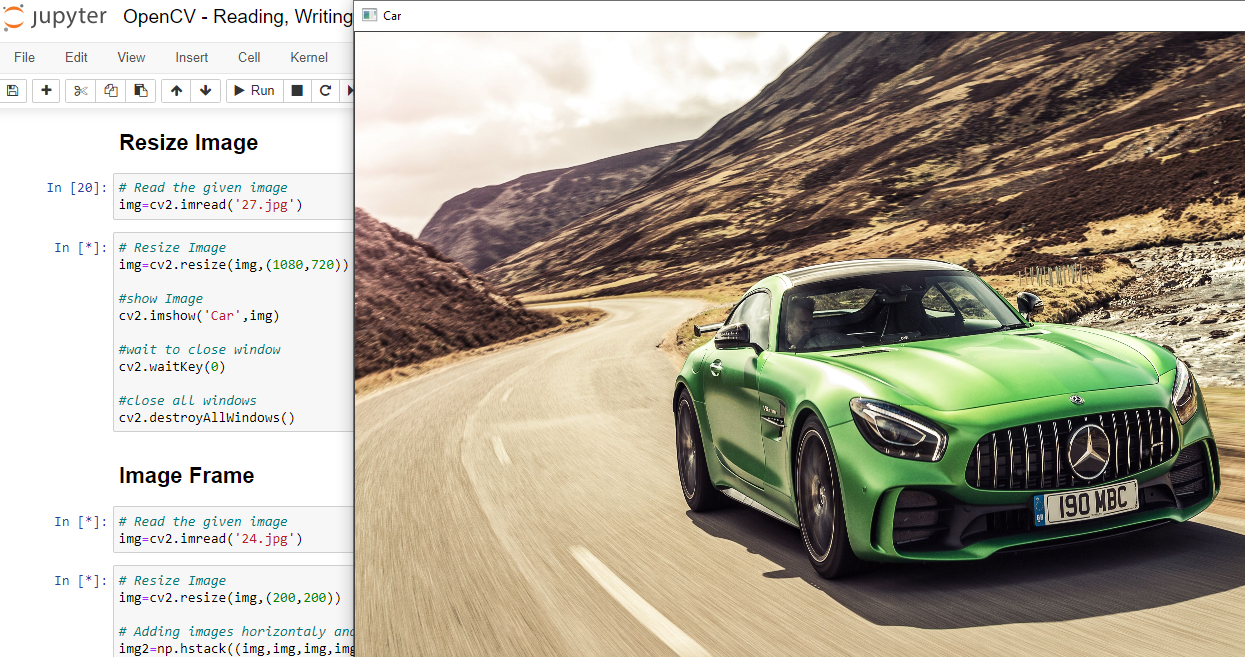
cv2.imshow('Car',img)

#wait to close window

cv2.waitKey(0)

#close all windows

cv2.destroyAllWindows()



## **Image Frame**

**Code:**

# Read the given image

img=cv2.imread('24.jpg')

# Resize Image

img=cv2.resize(img,(200,200))

# Adding images horizontally and Vertically

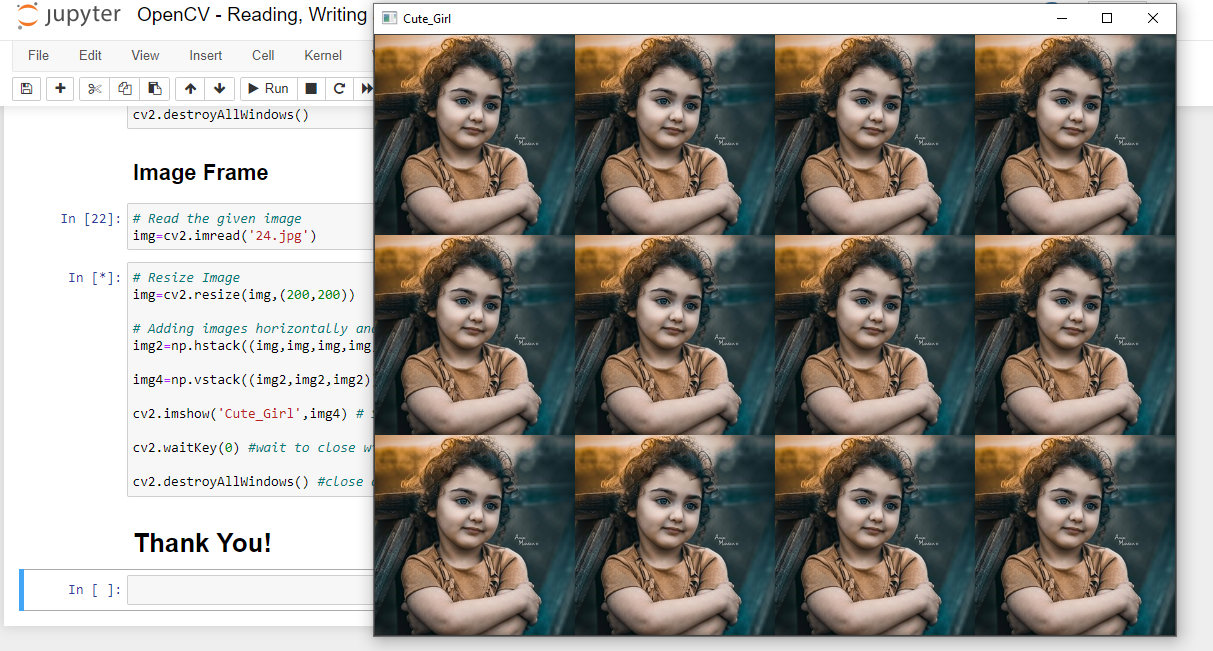
img2=np.hstack((img,img,img,img,)) # Same image add 4 times horizontally

img4=np.vstack((img2,img2,img2)) # horizontally image add 2 times vertically

cv2.imshow('Cute\_Girl',img4) # Show images

cv2.waitKey(0) #wait to close window

cv2.destroyAllWindows() #close all windows



**Blog Site:** <https://www.smiit.xyz/opencv-read-write-and-display-images/>

**GitHub Link:** <https://github.com/SMIIT-Projects/OpenCV-Reading-Writing-and-Displaying-Images>

## **Tools and Technologies:**

The Code is written in Python 3.8.5.

## **Used libraries:**

opencv-python==4.5.1.48

matplotlib==3.4.1

numpy==1.20.2

pandas==1.2.4

scipy==1.6.2

seaborn==0.11.1

Thank You!