

## Experiment 1 : Perform basic Image Handling and Image Processing operations on the image.

### Processing of image

#### Use

**Imread()**

**Resize()**

**Flip()**

**Colored to grayscale**

**Grayscale to colored**

**Imshow()**

**Print()**

**save the generated Image**

```
In [16]: import cv2 #import cv Libraries
```

```
In [17]: """
It contains two parameters
    1. Location or path
    2. Number to define what type of image you need.
"""
#Colored
Img1 = cv2.imread("Nature.png", 1)
```

```
In [18]: #Print image pixel array
print(Img1)
```

```

[[[248 248 248]
   [247 247 247]
   [247 247 247]
   ...
   [247 247 247]
   [247 247 247]
   [248 248 248]]

[[[247 247 247]
   [247 247 247]
   [247 247 247]
   ...
   [247 247 247]
   [247 247 247]
   [247 247 247]]

[[[247 247 247]
   [246 246 246]
   [246 246 246]
   ...
   [246 246 246]
   [246 246 246]
   [247 247 247]]

...

[[[247 247 247]
   [246 246 246]
   [246 246 246]
   ...
   [246 246 246]
   [246 246 246]
   [247 247 247]]

[[[247 247 247]
   [247 247 247]
   [247 247 247]
   ...
   [247 247 247]
   [247 247 247]
   [247 247 247]]

[[[248 248 248]
   [247 247 247]
   [247 247 247]
   ...
   [247 247 247]
   [247 247 247]
   [248 248 248]]]

```

```

In [19]: #Show Original Image
cv2.imshow("Nature.png",Img1)
cv2.waitKey(1)

```

```

Out[19]: -1

```

```

In [20]: #Grayscale
Img2 = cv2.imread("Nature.png", 0)

```

In [21]: `print(Img2)`

```
[[248 247 247 ... 247 247 248]
 [247 247 247 ... 247 247 247]
 [247 246 246 ... 246 246 247]
 ...
 [247 246 246 ... 246 246 247]
 [247 247 247 ... 247 247 247]
 [248 247 247 ... 247 247 248]]
```

In [22]: *#Show original Image*  
`cv2.imshow("Nature.png", Img2)`  
`cv2.waitKey(1)`

Out[22]: -1

In [23]: *#unchanged or improve saturation*  
`Img3 = cv2.imread("Nature.png", -1)` *#unchanged or improve saturation*

In [24]: *# to convert the image into array which is used to manipulate image*  
`print(Img3)`

```

[[[248 248 248]
  [247 247 247]
  [247 247 247]
  ...
  [247 247 247]
  [247 247 247]
  [248 248 248]]

[[[247 247 247]
  [247 247 247]
  [247 247 247]
  ...
  [247 247 247]
  [247 247 247]
  [247 247 247]]

[[[247 247 247]
  [246 246 246]
  [246 246 246]
  ...
  [246 246 246]
  [246 246 246]
  [247 247 247]]

...

[[[247 247 247]
  [246 246 246]
  [246 246 246]
  ...
  [246 246 246]
  [246 246 246]
  [247 247 247]]

[[[247 247 247]
  [247 247 247]
  [247 247 247]
  ...
  [247 247 247]
  [247 247 247]
  [247 247 247]]

[[[248 248 248]
  [247 247 247]
  [247 247 247]
  ...
  [247 247 247]
  [247 247 247]
  [248 248 248]]]

```

```

In [25]: #Show original Image
cv2.imshow("Nature.png",Img3)
cv2.waitKey(1)

```

```

Out[25]: -1

```

```

In [26]: """
resize function contains parameters as:
1. Image name

```

```
2. Width and height values
"""
Img1 = cv2.resize(Img1, (500,500))
```

```
In [27]: cv2.imshow("Resized Image is:",Img1)
cv2.waitKey(0) # to hold the output screen
cv2.destroyAllWindows() # to destroy or free memory
```

## Flip the image

### Parameters:

1. Name of the image
2. Value

Value can be 0, 1, -1

0 - upside down

1 - left points to right or right to left

-1 - combination of 0 and 1

```
In [28]: Img1=cv2.flip(Img1,0)
```

```
In [29]: cv2.imshow("Flip image" ,Img1)
cv2.waitKey(0)
```

```
Out[29]: -1
```

```
In [30]: #How to save any image
cv2.imwrite( "Nature.png", Img1)
```

```
Out[30]: True
```

```
In [31]: #Else:
cv2.destroyAllWindows()
```