

```
!pip install opencv-python
```

```
!pip install matplotlib
```

```
!pip install pandas
```

```
!pip install numpy
```

```
!pip install pillow
```

```
!pip install keras
```

```
!pip install tensorflow
```

```
!pip install scikit-learn
```

```
!pip install scikit-image
```

```
#Grayscale Image
```

```
import matplotlib.pyplot as plt
```

```
import cv2
```

```
image_path = "      "
```

```
img = cv2.imread(image_path)
```

```
print(img)
```

```
gray_img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

```
print(grey_img)
```

```
plt.figure(figsize=(8, 4))
```

```
plt.subplot(1, 2, 1)
```

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
plt.title('Original Image')
```

```
plt.axis("off")
```

```
plt.subplot(1, 2, 2)
```

```
plt.imshow(gray_img, cmap='gray')
```

```
plt.title('Grayscale Image')
```

```
plt.axis("off")
```

```
#Brightness
```

```
b_fact = 1.8
```

```
bright_img = cv2.convertScaleAbs(img, alpha=b_fact, beta=0)
```

```
plt.figure(figsize =(8,6))
```

```
plt.subplot(1,2,1)
```

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Original img")
```

```
plt.axis('off')
```

```
plt.subplot(1,2,2)
```

```
plt.imshow(cv2.cvtColor(bright_img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Bright img")
```

```
plt.axis('off')
```

```
#Contrast
```

```
import numpy as np
```

```
c_fact=1.7
```

```
const_img = cv2.addWeighted(img, c_fact, np.zeros_like(img),0,0)
```

```
plt.figure(figsize =(8,6))
```

```
plt.subplot(1,3,1)
```

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Original img")
```

```
plt.axis('off')
```

```
plt.subplot(1,3,2)
```

```
plt.imshow(cv2.cvtColor(bright_img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Bright img")
```

```
plt.axis('off')
```

```
plt.subplot(1,3,3)
```

```
plt.imshow(cv2.cvtColor(const_img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Weighted Image")
```

```
plt.axis('off')
```

#shadow

```
ret, mask = cv2.threshold(gray_img, 200, 255, cv2.THRESH_BINARY_INV)
```

```
rgba = cv2.cvtColor(img, cv2.COLOR_BGR2RGBA)
```

```
shadow = 0.9
```

```
rgba[:, :, 3] = cv2.bitwise_or(rgba[:, :, 3], mask * int(255 * shadow))
```

```
plt.figure(figsize =(8,6))
```

```
plt.subplot(1,2,1)
```

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Original img")
```

```
plt.axis('off')
```

```
plt.subplot(1,2,2)
```

```
plt.imshow(rgba)
```

```
plt.title("Shadowed img")
```

```
plt.axis('off')
```

#hue

```
hsv_image = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

```
hue_shift = 50
```

```
hsv_image[:, :, 0] = (hsv_image[:, :, 0] + hue_shift) % 180
```

```
modified_image = cv2.cvtColor(hsv_image, cv2.COLOR_HSV2BGR)
```

```
plt.figure(figsize=(10, 5))
```

```
plt.subplot(1, 2, 1)
```

```
plt.title('Original Image')
```

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
plt.axis('off')
```

```
plt.subplot(1, 2, 2)
```

```
plt.title('Modified Image')
```

```
plt.imshow(cv2.cvtColor(modified_image, cv2.COLOR_BGR2RGB))
```

```
plt.axis('off')
```

```
#temparature
```

```
from skimage import exposure
```

```
gamma_fact = 1.5
```

```
after_gama = exposure.adjust_gamma(const_img, gamma_fact)
```

```
plt.figure(figsize =(14,6))
```

```
plt.subplot(1,4,1)
```

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Original img")
```

```
plt.axis('off')
```

```
plt.subplot(1,4,2)
```

```
plt.imshow(cv2.cvtColor(bright_img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Bright Image")
```

```
plt.axis('off')
```

```
plt.subplot(1,4,3)
```

```
plt.imshow(cv2.cvtColor(const_img, cv2.COLOR_BGR2RGB))
```

```
plt.title("Weighted Image")
```

```
plt.axis('off')
```

```
plt.subplot(1,4,4)
```

```
plt.imshow(cv2.cvtColor(after_gama, cv2.COLOR_BGR2RGB))
```

```
plt.title("Gamma Image")
```

```
plt.axis('off')
```

```
=====
```

```
!pip install scikit-image
```

```
from skimage import exposure
```

```
#Gamma
```

```
G_fact = 1.5
```

```
G_image = exposure.adjust_gamma(const_img,G_fact)
```

```
#log image
```

```
L_fact = 1.2
```

```
Log_image = exposure.adjust_log(G_image, L_fact, inv = False)
```

```
#sigmoid image
```

```
S_fact = 1.5
```

```
S_image = exposure.adjust_sigmoid(Log_image, S_fact, inv=False)
```

```
plt.figure(figsize =(10,8))
```

```
plt.subplot(1,5,1)
```

```
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

plt.title("Original img")

plt.axis('off')

plt.subplot(1,5,2)

plt.imshow(cv2.cvtColor(const_img, cv2.COLOR_BGR2RGB))

plt.title("Weighted Image")

plt.axis('off')

plt.subplot(1,5,3)

plt.imshow(cv2.cvtColor(G_image, cv2.COLOR_BGR2RGB))

plt.title("Gamma Image")

plt.axis('off')

plt.subplot(1,5,4)

plt.imshow(cv2.cvtColor(G_image, cv2.COLOR_BGR2RGB))

plt.title("Log Image")

plt.axis('off')

plt.subplot(1,5,5)

plt.imshow(cv2.cvtColor(Log_image, cv2.COLOR_BGR2RGB))

plt.title("Sigmoid Image")

plt.axis('off')

#Cumulative img
cu_img= exposure.cumulative_distribution(img, nbins=256)
```



```
plt.figure(figsize =(8,6))
plt.subplot(1,1,1)
plt.imshow(cu_img)
plt.title("Cumulative img")
plt.axis('off')
```

#Equalize_adapthist img

```
ad_img = exposure.equalize_adapthist(img, kernel_size=(2,2), clip_limit=0.01, nbins=256)
```

```
plt.figure(figsize =(10,8))
plt.subplot(1,2,1)
plt.imshow(ad_img)
plt.title("Equalize_adapthist img")
plt.axis('off')
```

Add, Sub & Mul image

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
image_path1 = "C:/Users/Student/Downloads/dog.jpg"
image_path2 = "C:/Users/Student/Downloads/cat.jpg"
img1 = cv2.imread(image_path1)
img2 = cv2.imread(image_path2)
r_image1 = cv2.resize(img1, (300, 200))
r_image2 = cv2.resize(img2, (300, 200))
added_img = cv2.add(r_image1, r_image2)
sub_img = cv2.subtract(r_image1, r_image2)
mul_image = cv2.multiply(r_image1, r_image2)

plt.figure(figsize=(12, 6))
plt.subplot(2, 5, 1)
plt.imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))
plt.title('Image 1')
plt.axis('off')

plt.subplot(2, 5, 2)
plt.imshow(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB))
plt.title('Image 2')
plt.axis('off')
```

```
plt.subplot(2, 5, 3)
plt.imshow(cv2.cvtColor(added_img, cv2.COLOR_BGR2RGB))
plt.title('Added Image')
plt.axis('off')
```

```
plt.subplot(2, 5, 4)
plt.imshow(cv2.cvtColor(sub_img, cv2.COLOR_BGR2RGB))
plt.title('Subtracted Image')
plt.axis('off')
```

```
plt.subplot(2, 5, 5)
plt.imshow(cv2.cvtColor(mul_image, cv2.COLOR_BGR2RGB))
plt.title('Mulplied Image')
plt.axis('off')
```