

# COMPUTER VISION

## ASSIGNMENT-1

AMOGH GARG – 2020UCO1688

---

Importing Libraries

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Task-1 : Read an image into a variable

Task-2 : Display that image

```
img = cv2.imread("Img-1.jpg",cv2.IMREAD_COLOR)
img = cv2.resize(img, (700,700))
cv2.imshow('Colored Image of Butterfly', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Task-3 : Convert the image into grayscale

```
img_gs = cv2.imread("Img-1.jpg",cv2.IMREAD_GRAYSCALE)
img_gs = cv2.resize(img_gs, (700,700))
cv2.imshow('Grayscale Image of Butterfly', img_gs)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Task-4 : Check the height and width of that image

```
img_gs.shape

(700, 700)
```

Task-5 : Extract RGB channels from the image

```
# extract red channel
red_channel = img[:, :, 2]

# create empty image with same shape as that of src image
red_img = np.zeros(img.shape)

#assign the red channel of src to empty image
red_img[:, :, 2] = red_channel

#save image
cv2.imwrite('Red_Img-1.jpg', red_img)

True

# extract green channel
green_channel = img[:, :, 1]

# create empty image with same shape as that of src image
green_img = np.zeros(img.shape)

#assign the red channel of src to empty image
green_img[:, :, 1] = green_channel
```

```

#save image
cv2.imwrite('Green_Img-1.jpg',green_img)

True

# extract blue channel
blue_channel = img[:, :, 0]

# create empty image with same shape as that of src image
blue_img = np.zeros(img.shape)

#assign the red channel of src to empty image
blue_img[:, :, 0] = blue_channel

#save image
cv2.imwrite('Blue_Img-1.jpg',blue_img)

True

```

Task-6 : Extract middle 100 pixels from the image

```

# Extracting the middle hundred pixels
cropped_img = img[300:400,300:400]
cv2.imshow('Center Image', cropped_img)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Task-7 : For the image, create one more image. Keep every 10th pixel in the horizontal direction and 20th pixel in vertical direction. Display the output matrix in image form.

```

# Extracting every 10th pixel in horizontal and every 20th pixel in vertical direction
modified_img = img[:, ::10, ::20]
cv2.imshow('Modified Image', modified_img)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Task-8 : Flip the image vertically

```

flipped_img = cv2.flip(img,0)
cv2.imshow('Flipped Image', flipped_img)
cv2.waitKey(0)
cv2.destroyAllWindows()

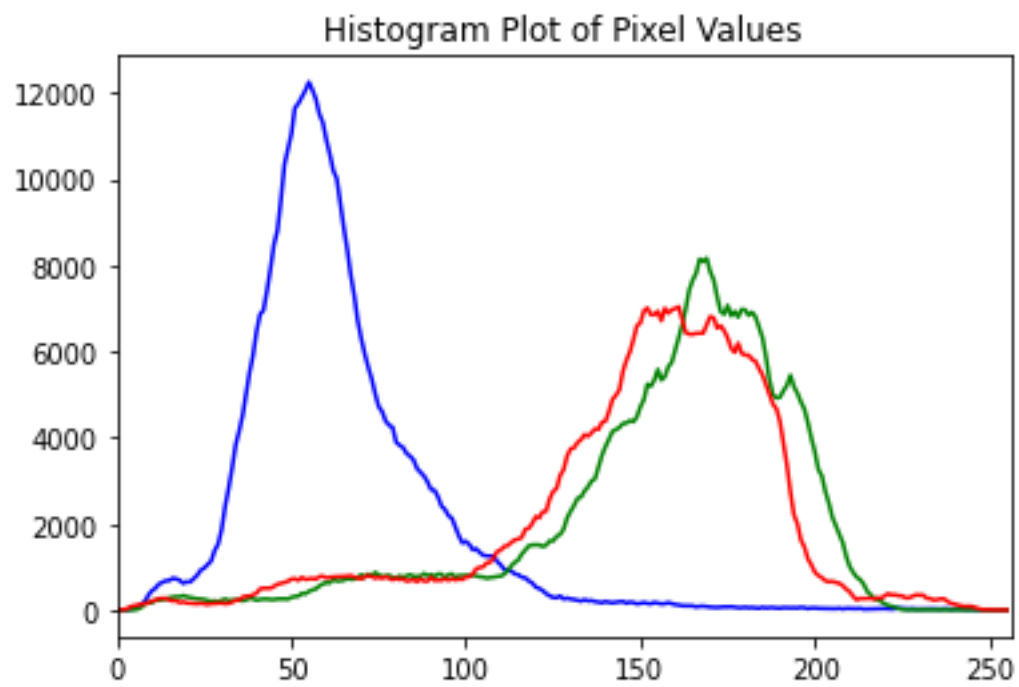
```

Task-9 : Draw the histogram plot of pixel values.

```

color = ('b', 'g', 'r')
for i,col in enumerate(color):
    histr = cv2.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.xlim([0,256])
plt.title('Histogram Plot of Pixel Values')
plt.show()

```



OUTPUT IMAGES:

