Ex. No. 01 - Image-Handling-and-Pixel-Transformations-Using-OpenCV

AIM:

Write a Python program using OpenCV that performs the following tasks:

1. Read and Display an Image.
2. Adjust the brightness of an image.
3. Modify the image contrast.
4. Generate a third image using bitwise operations.

Software Required:

- Anaconda - Python 3.7
- Jupyter Notebook (for interactive development and execution)

Algorithm:

Step 1:

Load an image from your local directory and display it.

Step 2:

Create a matrix of ones (with data type float64) to adjust brightness.

Step 3:

Create brighter and darker images by adding and subtracting the matrix from the original image.
Display the original, brighter, and darker images.

Step 4:

Modify the image contrast by creating two higher contrast images using scaling factors of 1.1 and 1.2 (without overflow fix).
Display the original, lower contrast, and higher contrast images.

Step 5:

Split the image (boy.jpg) into B, G, R components and display the channel

PROGRAM DEVELOPED BY:

```
NAME : RAJU D
REG NO: 212224240128
```

1. Read the image ('Eagle_in_Flight.jpg') using OpenCV imread() as a grayscale image.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
img =cv2.imread('Eagle_in_Flight.jpg',cv2.IMREAD_COLOR)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

2. Print the image width, height & Channel.

```
img.shape
```

3. Display the image using matplotlib imshow().

```
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2GRAY)
plt.imshow(img_gray,cmap='grey')
plt.show()
```

4. Save the image as a PNG file using OpenCV imwrite().

```
img=cv2.imread('Eagle_in_Flight.jpg')
cv2.imwrite('Eagle.png',img)
```

5. Read the saved image above as a color image using cv2.cvtColor().

```
img=cv2.imread('Eagle_in_Flight.jpg')
cv2.imwrite('Eagle.png',img)
```

6. Display the Colour image using matplotlib imshow() & Print the image width, height & channel.

```
plt.imshow(img)
plt.show()
img.shape
```

7. Crop the image to extract any specific (Eagle alone) object from the image.

```
crop = img_rgb[0:450,200:550]
plt.imshow(crop[:,:,::-1])
plt.title("Cropped Region")
plt.axis("off")
plt.show()
crop.shape
```

8. Resize the image up by a factor of 2x.

```
res= cv2.resize(crop,(200*2, 200*2))
```

9. Flip the cropped/resized image horizontally.

```
flip= cv2.flip(res,1)
plt.imshow(flip[:,:,::-1])
plt.title("Flipped Horizontally")
plt.axis("off")
```

10. Read in the image ('Apollo-11-launch.jpg').

```
img=cv2.imread('Apollo-11-launch.jpg',cv2.IMREAD_COLOR)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img_rgb.shape
```

11. Add the following text to the dark area at the bottom of the image (centered on the image):

```
text = cv2.putText(img_rgb, "Apollo 11 Saturn V Launch, July 16, 1969",
(300, 700),cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
plt.imshow(text, cmap='gray')
plt.title("New image")
plt.show()
```

12. Draw a magenta rectangle that encompasses the launch tower and the rocket.

```
rcol= (255, 0, 255)
cv2.rectangle(img_rgb, (400, 100), (800, 650), rcol, 3)
```

13. Display the final annotated image.

```
plt.title("Annotated image")
plt.imshow(img_rgb)
plt.show()
```

14. Read the image ('Boy.jpg').

```
img =cv2.imread('boy.jpg',cv2.IMREAD_COLOR)
img_rgb= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

15. Adjust the brightness of the image.

```
m = np.ones(img_rgb.shape, dtype="uint8") * 50
```

16. Create brighter and darker images.

```
img_brighter = cv2.add(img_rgb, m)
img_darker = cv2.subtract(img_rgb, m)
```

17. Display the images (Original Image, Darker Image, Brighter Image).

```
plt.figure(figsize=(10,5))
plt.subplot(1,3,1), plt.imshow(img_rgb), plt.title("Original Image"),
plt.axis("off")
plt.subplot(1,3,2), plt.imshow(img_brighter), plt.title("Brighter
Image"), plt.axis("off")
```

```
plt.subplot(1,3,3), plt.imshow(img_darker), plt.title("Darker Image"),
plt.axis("off")
plt.show()
```

18. Modify the image contrast.

```
matrix1 = np.ones(img_rgb.shape, dtype="float32") * 1.1
matrix2 = np.ones(img_rgb.shape, dtype="float32") * 1.2
img_higher1 = cv2.multiply(img.astype("float32"),
matrix1).clip(0,255).astype("uint8")
img_higher2 = cv2.multiply(img.astype("float32"),
matrix2).clip(0,255).astype("uint8")
```

19. Display the images (Original, Lower Contrast, Higher Contrast).

```
plt.figure(figsize=(10,5))
plt.subplot(1,3,1), plt.imshow(img), plt.title("Original Image"),
plt.axis("off")
plt.subplot(1,3,2), plt.imshow(img_higher1), plt.title("Higher Contrast
(1.1x)"), plt.axis("off")
plt.subplot(1,3,3), plt.imshow(img_higher2), plt.title("Higher Contrast
(1.2x)"), plt.axis("off")
plt.show()
```

20. Split the image (boy.jpg) into the B,G,R components & Display the channels.

```
b, g, r = cv2.split(img)
plt.figure(figsize=(10,5))
plt.subplot(1,3,1), plt.imshow(b, cmap='gray'), plt.title("Blue
Channel"), plt.axis("off")
plt.subplot(1,3,2), plt.imshow(g, cmap='gray'), plt.title("Green
Channel"), plt.axis("off")
plt.subplot(1,3,3), plt.imshow(r, cmap='gray'), plt.title("Red Channel"),
plt.axis("off")
plt.show()
```

21. Merged the R, G, B , displays along with the original image

```
merged_rgb = cv2.merge([r, g, b])
plt.figure(figsize=(5,5))
plt.imshow(merged_rgb)
plt.title("Merged RGB Image")
plt.axis("off")
plt.show()
```

22. Split the image into the H, S, V components & Display the channels.

```
hsv_img = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
h, s, v = cv2.split(hsv_img)
plt.figure(figsize=(10,5))
plt.subplot(1,3,1), plt.imshow(h, cmap='gray'), plt.title("Hue Channel"),
plt.axis("off")
plt.subplot(1,3,2), plt.imshow(s, cmap='gray'), plt.title("Saturation
Channel"), plt.axis("off")
plt.subplot(1,3,3), plt.imshow(v, cmap='gray'), plt.title("Value
Channel"), plt.axis("off")
plt.show()
```

23. Merged the H, S, V, displays along with original image.

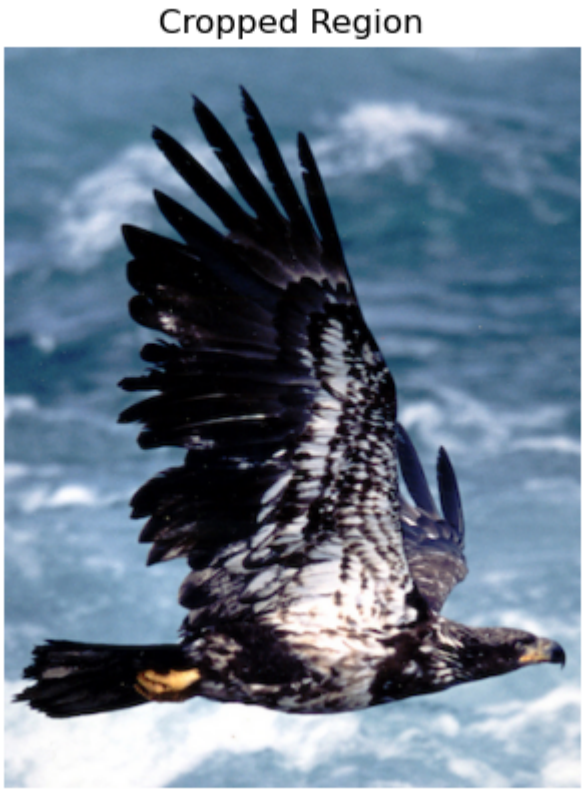
```
merged_hsv = cv2.cvtColor(cv2.merge([h, s, v]), cv2.COLOR_HSV2RGB)
combined = np.concatenate((img_rgb, merged_hsv), axis=1)
plt.figure(figsize=(10, 5))
plt.imshow(combined)
plt.title("Original Image  &  Merged HSV Image")
plt.axis("off")
plt.show()
```

Output:

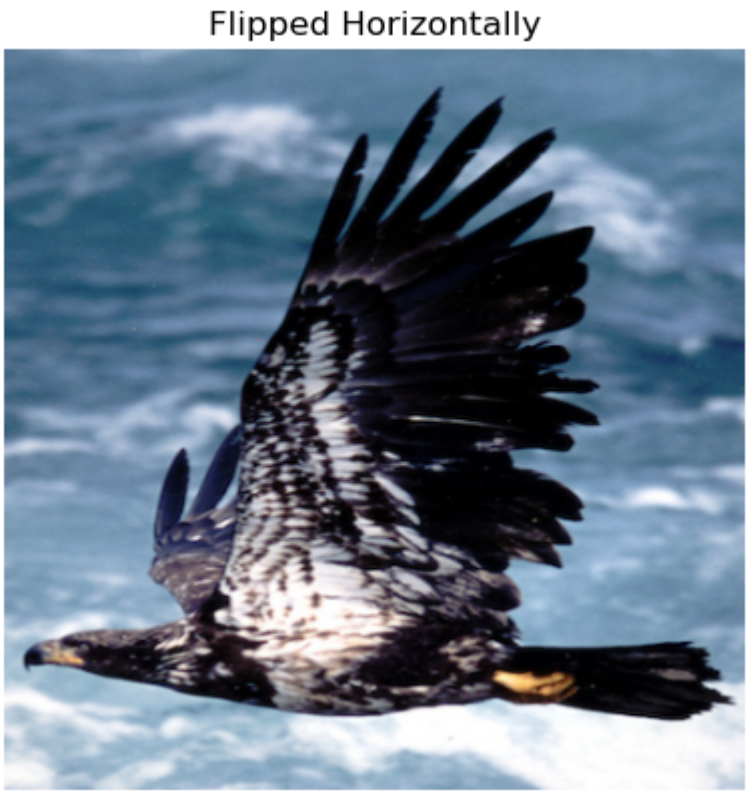
i) Read and Display an Image. 1.Read 'Eagle_in_Flight.jpg' as grayscale and display:

2.Save image as PNG and display:

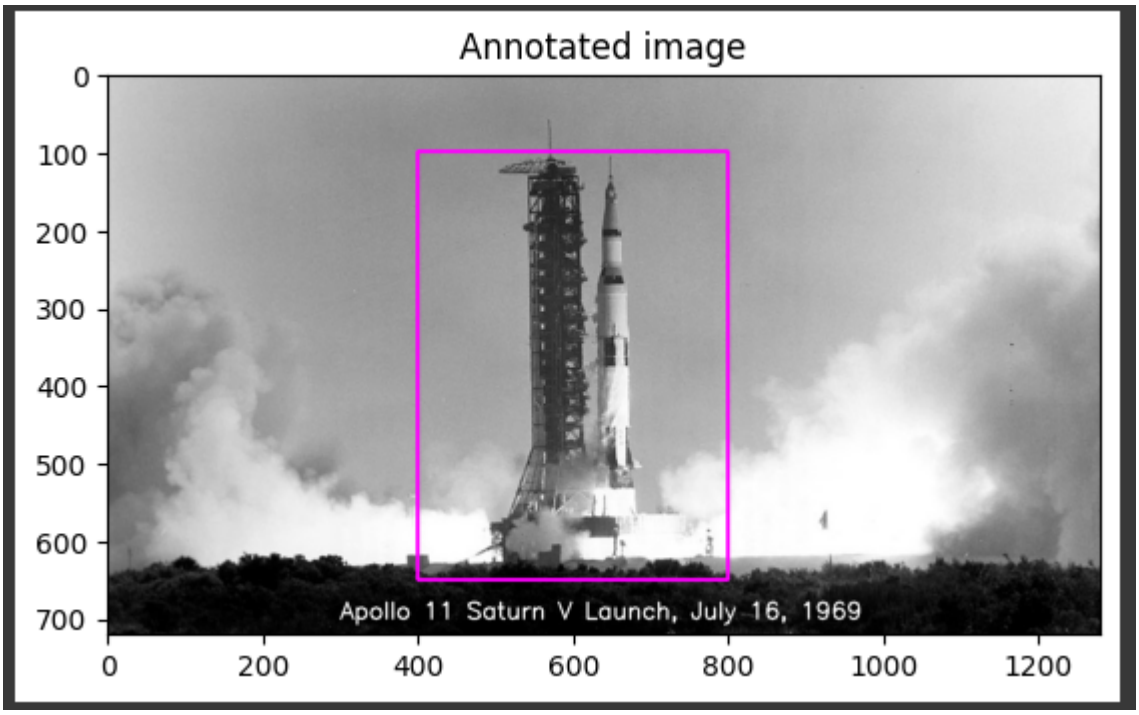
3.Cropped image



4.Resize and flip Horizontally:



5.Read 'Apollo-11-launch.jpg' and Display the final annotated image:



*ii)** Adjust Image Brightness.

1.Create brighter and darker images and display:

iii) Modify Image Contrast.

1.Modify contrast using scaling factors 1.1 and 1.2

iv) Generate Third Image Using Bitwise Operations.

1.Split 'Boy.jpg' into B, G, R components and display:

2.Merge the R, G, B channels and display:

3.Split the image into H, S, V components and display:

4.Merge the H, S, V channels and display:

Result:

Thus, the images were read, displayed, brightness and contrast adjustments were made, and bitwise operations were performed successfully using the Python program.