

Quiz 05

Name: Muhammad Arqam Nisar

Reg: 318376

- **Load an image using opencv library**
- **Convert the image into numpy array**
- **Apply reshaping**
- **Save the image**
- **Apply 5 different functions using numpy and opencv**

Loading an image:

```
import cv2

import numpy as np

import matplotlib.pyplot as plt

#Loading an image:

img = cv2.imread("9.jpg", 0)

cv2.imshow("Image", img)

cv2.waitKey(0)
```

Converting into array:

```
#Converting image to array:

im = np.array(img)

im.shape
```

Reshaping:

```
#Reshaping into another image:

im1 = np.reshape(im, (1618,640))

cv2.imshow("Image reshaped", im1)

cv2.waitKey(0)
```

Saving the image as new file:

```
#Saving the new image:

cv2.imwrite('lion_resized.jpg', im1)
```

```
#Reloading the new image from the current directory:
```

```
new_img = cv2.imread('lion_resized.jpg', 0)
```

Numpy functions on Image:

```
#NumPy Functions:
```

```
#Function 01(Resizing the image):
```

```
resized_image = np.resize(img, (400, 2560))
```

```
#Function 02(Flipping the image):
```

```
Flipped_image = np.flip(img)
```

```
#Function 03(Negative of an image):
```

```
image = 255 - img
```

```
#Function 04(Rotate image):
```

```
rotated_image = np.rot90(img)
```

```
#Function 05(Padding):
```

```
padded_image = np.pad(img, ((150, 150), (120, 120)))
```

```
#Showing all the results alongside original image, of the functions applied:
```

```
plt.subplot(241),plt.imshow(img,'gray'),plt.title('Original')
```

```
plt.subplot(242),plt.imshow(new_img,'gray'),plt.title('Resized')
```

```
plt.subplot(243),plt.imshow(resized_image,'gray'),plt.title('Resized')
```

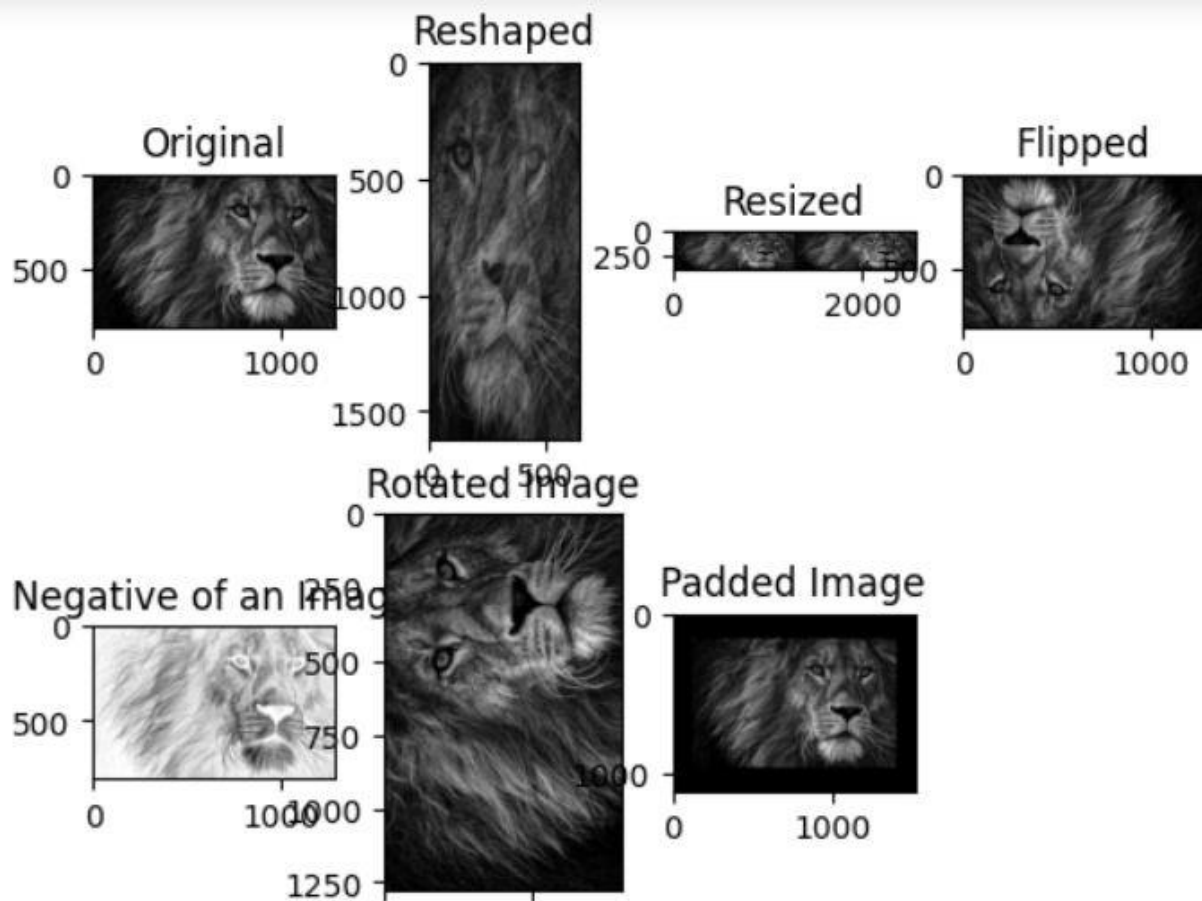
```
plt.subplot(244),plt.imshow(Flipped_image,'gray'),plt.title('Flipped')
```

```
plt.subplot(245),plt.imshow(image,'gray'),plt.title('Negative of an Image')
```

```
plt.subplot(246),plt.imshow(rotated_image,'gray'),plt.title('Rotated Image')
```

```
plt.subplot(247),plt.imshow(padded_image,'gray'),plt.title('Padded Image')
```

Output:



OpenCV functions on Image:

#OpenCV functions on the image:

#Function 01(Rotation):

```
rotated_image_cv = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
```

#Function 02(Flipping):

```
flipped_image_cv = cv2.flip(img, 0)
```

#Function 03(Blurred Image):

```
blurred_image_cv = cv2.blur(img, (30, 30))
```

#Function 04(Canny Edge Detection):

```
canny_image_cv = cv2.Canny(img, 50, 120)
```

#Function 05(Pyramid Upsampling):

```
pyramid_image_cv = cv2.pyrUp(img)
```

#Showing all the results alongside original image, of the functions applied:

```
plt.subplot(231),plt.imshow(img,'gray'),plt.title('Original')
```

```
plt.subplot(232),plt.imshow(rotated_image_cv,'gray'),plt.title('Rotated Image')
```

```
plt.subplot(233),plt.imshow(flipped_image_cv,'gray'),plt.title('Flipped Image')
```

```
plt.subplot(234),plt.imshow(blurred_image_cv,'gray'),plt.title('Blurry Image')
```

```
plt.subplot(235),plt.imshow(canny_image_cv,'gray'),plt.title('Canny Image')
```

```
plt.subplot(236),plt.imshow(pyramid_image_cv,'gray'),plt.title('Pyramid Image')
```

Output:

