E/16/134
Herath H.M.Y.B

CO543 – Image Processing Lab04

Task 1
1) Use of Second Derivative for Image Enhancement: The Laplacian

● Write a program to implement "The Laplacian" and note the effects on the given image.

Input image

Output image

Function: laplacian_filter = laplacian(img1)

2) Use of First Derivative for Image Enhancement: The Gradient

● Write a program to implement "Robert Cross Gradient Operator" and observe the changes in the image.

Input image

Output image

Function: robert_filter = RCG(img1)

● Write a program to implement “Sobel Operators” and observe the changes on the image.

Input image

Output image

Function: sobel_car = Sobel(img1)

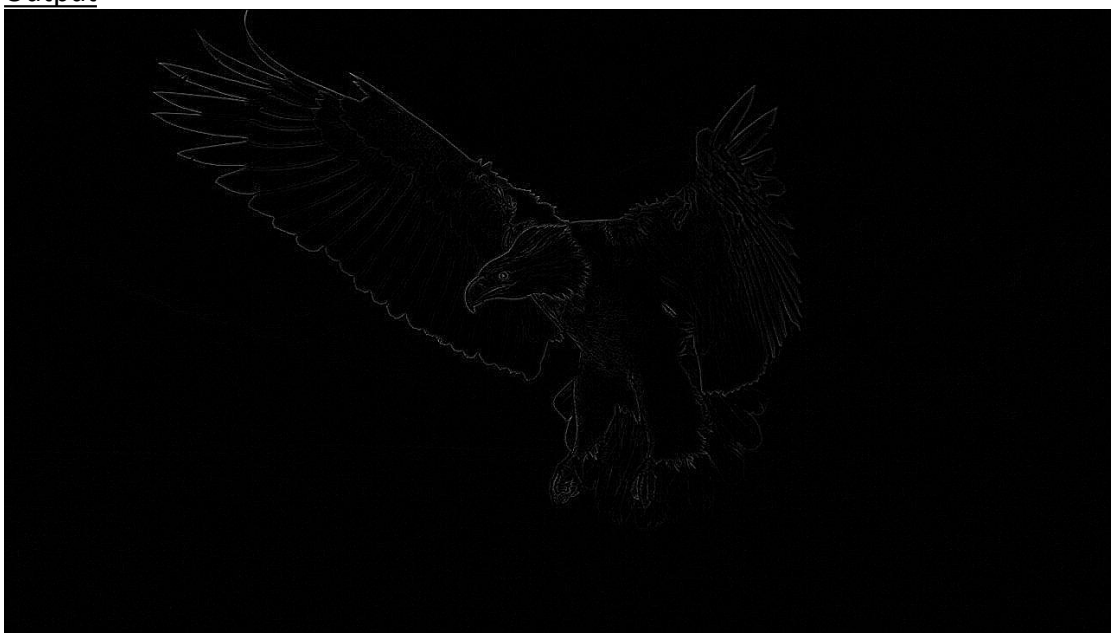
Task 2

1. Read a given image.

Image Read(Input)

2. Take the Laplacian of the image using the Second Derivative.

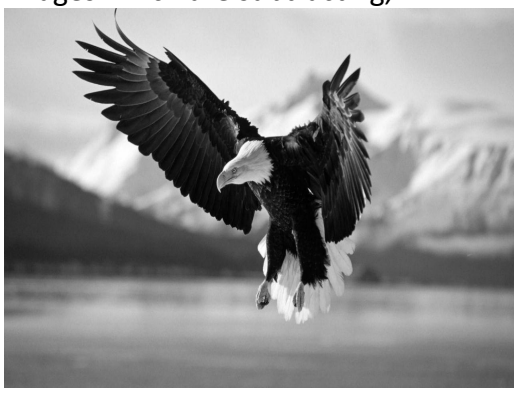
Output



3. Sharpen image by adding original image and laplacian image.

In this images are subtracted because middle coefficient of laplacian mask is negative.

Images which are subtracting,



Output image

Function: sharpened_img = cv2.subtract(img, laplacian_filter)

4. Take the Sobel gradient of the image using the First Derivative.

Input image

Output image

Function: sobel_img = Sobel(img)

5. Smooth Sobel image using 5× 5averaging filter.

Input image

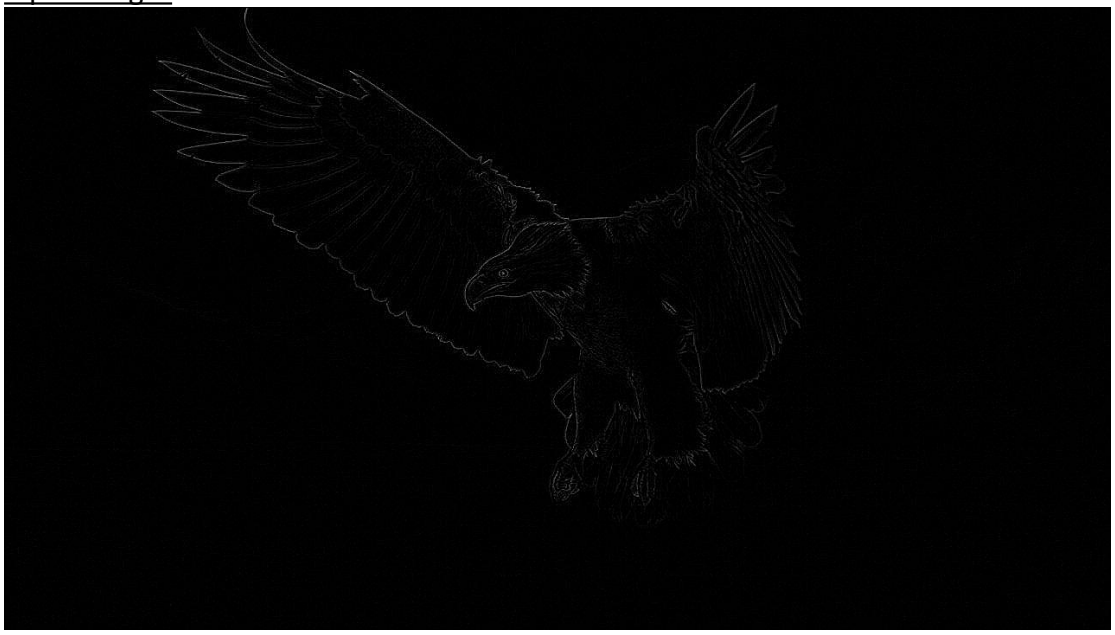
Output image

Function: sobel_img = Sobel(img)

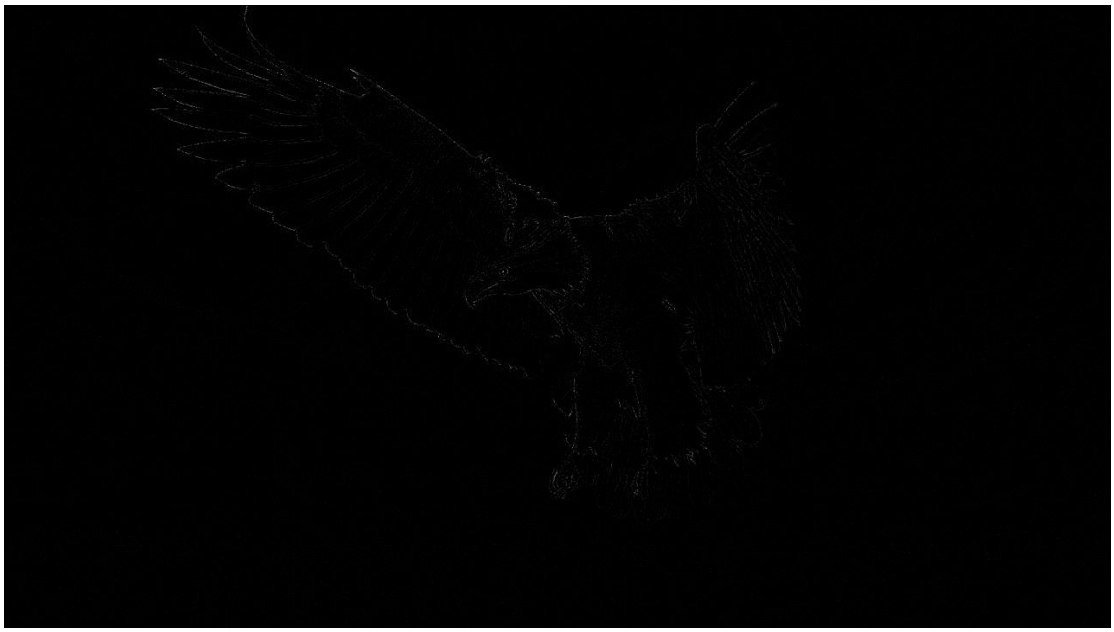
6. Mask the laplacian image using smooth Sobel image.

Bitwise And is used.

Input images



Output image

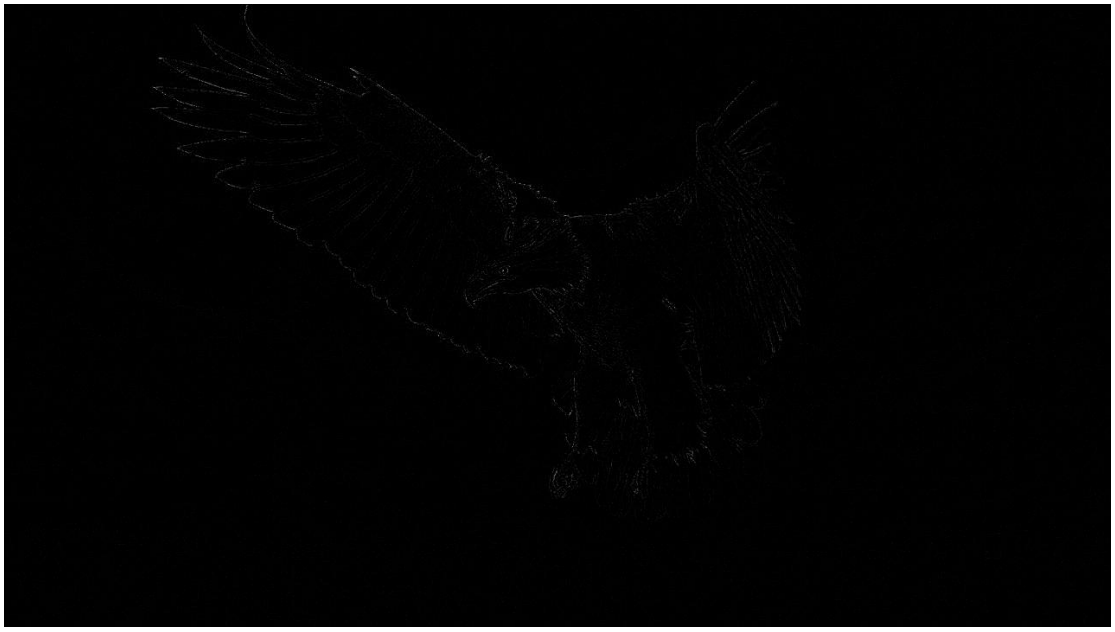


Function: masked_img = cv2.bitwise_and(smooth_sobel_img, laplacian_filter)

7. Add product image with the original image.

In this also subtraction is done due to negative middle coefficient in laplace operation.

Input images



Output image

Function: added_img = cv2.subtract(img, masked_img)

8. For better result apply power_law transformation

Input image

Output image

Function: final = powerTrans(added_img, 0.6)