AD18511 – DEEP LEARNING LABORATORY

DATE:

EX.NO: 7 **KERNEL IMPLEMENTATION**

AIM:

To implement different kernels and perform convolution on an image using CNN.

DESCRIPTION:

- In image processing, a kernel, convolution matrix, or mask is a small matrix used for blurring, sharpening, embossing, edge detection, and more.
- This is accomplished by doing a convolution between the kernel and an image.
- When each pixel in the output image is a function of the nearby pixels (including itself) in the input image, the kernel is that function.

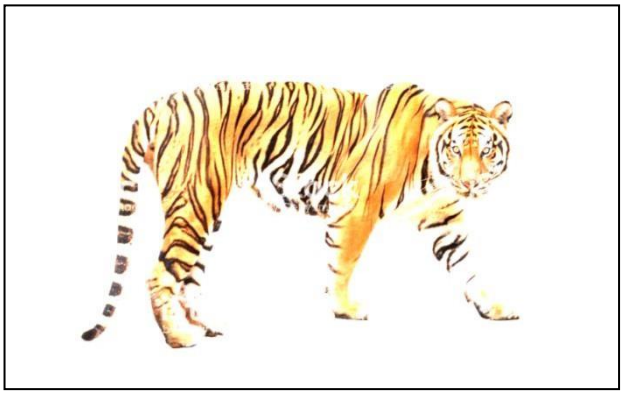
PROGRAM:

```
#Grascale image vector 0-255 pixels(2D)

import cv2
import numpy as np

#load an image
image=cv2.imread('/home/user/Desktop/pro.jpg')

#Define different kernels
kernel_identity=np.array([[0,0,0], [0,1,0], [0,0,0]])

kernel_edge_detection=np.array([[-1,-1,-1],[-1,8,-1],  [-1,-1,-1]])
kernel_sharpen=np.array([[0,-1,0], [-1,5,-1], [0,-1,0]])
kernel_blur=np.array([[1,2,1],[2,4,2], [1,2,1]])/16.0

#Apply kernels to the image
result_identity=cv2.filter2D(image,-1,kernel_identity)
result_edge_detection=cv2.filter2D(image,-1,kernel_edge_detection)
result_sharpen=cv2.filter2D(image,-1,kernel_sharpen)
result_blur=cv2.filter2D(image,-1,kernel_blur)

#Display the original and processed image
cv2.imshow('Original image',image)
cv2.imshow('Identity Kernel',result_identity)
cv2.imshow('Edge Detection Kernel',result_edge_detection)
cv2.imshow('Sharpen Kernel',result_sharpen)
cv2.imshow('Blur Kernel',result_blur)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

OUTPUT:

Original

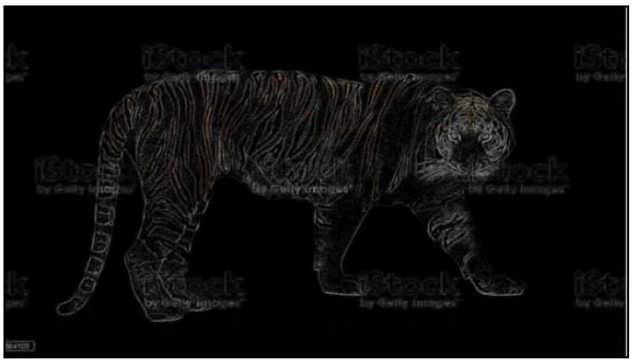
Identity Kernel



Sharpen

Edge detection

Blur Image



RESULT:

The kernels are created and applied to the image and the result is displayed successfully.