1)

import cv2

import numpy as np

kernel = np.ones((5,5),np.uint8) print(kernel)

path = "C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/cat.jpeg" img =cv2.imread(path)

imgGray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY) cv2.imshow("GrayScale",imgGray)

cv2.waitKey(0)

2)

import cv2

import numpy as np

kernel = np.ones((5,5),np.uint8) print(kernel)

path = "C:/Users/divya/OneDrive/Documents/COMPUTER VISION/Picture2.jpg" img =cv2.imread(path)

imgGray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY) imgBlur = cv2.GaussianBlur(imgGray,(7,7),0)

cv2.imshow("Img Blur",imgBlur) cv2.waitKey(0)



3)

import cv2

import numpy as np

kernel = np.ones((5,5),np.uint8) print(kernel)

path = "C:/Users/divya/OneDrive/Documents/COMPUTER VISION/CANNY.jpg" img =cv2.imread(path)

imgGray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY) imgBlur = cv2.GaussianBlur(imgGray,(7,7),0)

imgCanny = cv2.Canny(imgBlur,100,200) cv2.imshow("Img Canny",imgCanny)



cv2.waitKey(0)

4)

import cv2

import numpy as np

image\_path ="E:/Computer Vision/computer vision input and output/4.Image using Dilate function input.png"

image = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

cv2.imshow("original image",image)

cv2.waitKey(0)

kernel = np.ones((5, 5), np.uint8)

dilated\_image = cv2.dilate(image, kernel, iterations=1)

cv2.imshow('Dilated Image', dilated\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

5)

import cv2

import numpy as np

kernel = np.ones((5,5),np.uint8) print(kernel)

path = "C:/Users/divya/OneDrive/Documents/COMPUTER VISION/erosion.jpg" img =cv2.imread(path)

imgGray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY) imgBlur = cv2.GaussianBlur(imgGray,(7,7),0)

imgCanny = cv2.Canny(imgBlur,100,200)

imgDilation = cv2.dilate(imgCanny,kernel , iterations = 10) imgEroded = cv2.erode(imgDilation,kernel,iterations=2)



cv2.imshow("Img Erosion",imgEroded) cv2.waitKey(0)

6)

import cv2

import numpy as np

cap = cv2.VideoCapture("C:/Users/divya/OneDrive/Documents/COMPUTER VISION/13 REASONS WHY")

if (cap.isOpened()== False):

print("Error opening video file") while(cap.isOpened()):

ret, frame = cap.read()

if ret == True:

cv2.imshow('Frame',frame)

if cv2.waitKey(250) & 0xFF == ord('q'): break

else:

break

cap.release()

8)

import cv2

import numpy as np

kernel = np.ones((5,5),np.uint8)

img = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/cat.jpeg",cv2.IMREAD\_COLOR)

img = cv2.resize(img,(600,600))

cv2.imshow("Output image",img) cv2.waitKey(0)

9)

import cv2

path ="C:/Users/Welcome/OneDrive/Pictures/SavedPictures/cat.jpeg"

src = cv2.imread(path)

window\_name = 'Image'

image = cv2.rotate(src, cv2.ROTATE\_180)

cv2.imshow(window\_name, image) cv2.waitKey(0)

10)

import cv2

import numpy as np

image = cv2.imread("E:/Computer Vision/computer vision input and output/8.scaling an image input.png")

if image is not None:

tx = 50

ty = 30

translation\_matrix = np.float32([[1, 0, tx], [0, 1, ty]])

translated\_image = cv2.warpAffine(image, translation\_matrix, (image.shape[1], image.shape[0]))

cv2.imshow("Original Image", image)

cv2.imshow("Translated Image", translated\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

else:

print("Error: Could not load the image.")

11)

import cv2

import numpy as np

img = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/afiine.jpg")

rows,cols,ch = img.shape

pts1 = np.float32([[50,50],[200,50],[50,200]])

pts2 = np.float32([[10,100],[200,50],[100,250]])

Matrix M = cv2.getAffineTransform(pts1,pts2)

dst = cv2.warpAffine(img,M,(cols,rows))

cv2.imshow("Affine Transform", dst)

cv2.waitKey(0)

cv2.destroyAllWindows()

12)

import cv2

import numpy as np

img = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/afiine.jpg")

image array rows,cols,ch = img.shape

pts1 = np.float32([[56,65],[368,52],[28,387],[389,390]])

pts2 = np.float32([[100,50],[300,0],[0,300],[300,300]])

M = cv2.getPerspectiveTransform(pts1,pts2)

transform matrix dst = cv2.warpPerspective(img,M,(cols, rows))

cv2.imshow('Transformed Image', dst)

cv2.waitKey(0)

cv2.destroyAllWindows()

14)

import cv2

import numpy as np

im\_src = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/afiine.jpg”)

pts\_src = np.array([[141, 131], [480, 159], [493, 630],[64, 601]])

im\_dst = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/afiine.jpg")

pts\_dst = np.array([[318, 256],[534, 372],[316, 670],[73, 473]])

h, status = cv2.findHomography(pts\_src, pts\_dst)

im\_out = cv2.warpPerspective(im\_src, h, (im\_dst.shape[1],im\_dst.shape[0]))

cv2.imshow("Source Image", im\_src)

cv2.imshow("Destination Image", im\_dst)

cv2.imshow("Warped Source Image", im\_out)

cv2.waitKey(0)

15)

import cv2

import numpy as np # Load images

img1 = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/afiine.jpg") img2 = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/cat.jpeg")

pts1 = np.array([[50, 50], [200, 50], [50, 200], [200, 200]])

pts2 = np.array([[100, 100], [300, 100], [100, 300], [300, 300]])

matrix using DLT H, \_ = cv2.findHomography(pts1, pts2)

dst = cv2.warpPerspective(img1, H, (img2.shape[1], img2.shape[0])) cv2.imshow('img1', img1) cv2.imshow('img2', img2)

cv2.imshow('dst', dst)

cv2.waitKey(0)

cv2.destroyAllWindows()

16)

import cv2

img = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/cat.jpeg")

cv2.imshow('Original', img)

cv2.waitKey(0)

img\_gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

img\_blur = cv2.GaussianBlur(img\_gray, (3,3), 0)

edges = cv2.Canny(image=img\_blur, threshold1=100, threshold2=200)

cv2.imshow('Canny Edge Detection', edges)

cv2.waitKey(0)

cv2.destroyAllWindows()

17)

import cv2

img = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/cat.jpeg")

cv2.imshow('Original', img)

cv2.waitKey(0)

img\_gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

img\_blur = cv2.GaussianBlur(img\_gray, (3,3), 0)

sobelx = cv2.Sobel(src=img\_blur, ddepth=cv2.CV\_64F, dx=1, dy=0, ksize=5)

cv2.imshow('Sobel X', sobelx) cv2.waitKey(0)

18)

import cv2

img = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/cat.jpeg")

img\_gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

img\_blur=cv2.GaussianBlur(img\_gray, (3,3), 0)

sobely = cv2.Sobel(src=img\_blur, ddepth=cv2.CV\_64F, dx=0, dy=1, ksize=5)

Detection Images cv2.imshow('Sobel Y', sobely)

19)

import cv2

img = cv2.imread("C:/Users/Welcome/OneDrive/Pictures/Saved Pictures/cat.jpeg”)

cv2.imshow('Original', img) cv2.waitKey(0)

img\_gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

img\_blur = cv2.GaussianBlur(img\_gray, (3,3), 0)

sobelxy = cv2.Sobel(src=img\_blur, ddepth=cv2.CV\_64F, dx=1, dy=1, ksize=5)

cv2.imshow('Sobel X Y using Sobel() function', sobelxy)

cv2.waitKey(0)

20)

import cv2

import numpy as np

img = cv2.imread("C:/Users/DIVYA/OneDrive/Pictures/ss.png")

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

kernel = np.array([[0,1,0], [1,-8,1], [0,1,0]])

sharpened = cv2.filter2D(gray,-1, kernel)

cv2.imshow('Original', gray)

cv2.imshow('Sharpened', sharpened)

cv2.waitKey(0)

cv2.destroyAllWindows()

21)

import cv2

import numpy as np

img = cv2.imread("C:/Users/Dama Prasoona/OneDrive/Pictures/21.png") gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

kernel = np.array([[0,1,0], [1,-4,1], [0,1,0]])

sharpened = cv2.filter2D(gray,-1, kernel)

cv2.imshow('Original', gray)

cv2.imshow('Sharpened', sharpened)

cv2.waitKey(0)

cv2.destroyAllWindows()

25)

a=imread("C:/Users/divya/Downloads/Girl with a Cat.png")

Lap=[0, 1, 0, 1,-4, 1, 0, 1, 0]

a1=conv2(a,Lap,"C:/Users/divya/Downloads/Girl with a Cat.png")

a2=uint8(a1)

imtool(abs(a-a2),[])

lap=[-1 ,-1,-1,-1, 8,-1,-1,-1 ,-1]

a3=conv2(a,lap,"C:/Users/divya/Downloads/Girl with a Cat.png")

a4=uint8(a3)

imtool(abs(a+a4),[])

26)

import cv2

img = cv2.imread("C:/Users/divya/Downloads/Girl with a Cat.png")

wm = cv2.imread("C:/Users/divya/OneDrive/Pictures/Saved Pictures/logo.jfif") h\_wm, w\_wm = wm.shape[:2]

h\_img, w\_img = img.shape[:2] center\_x = int(w\_img/2) center\_y = int(h\_img/2)

top\_y = center\_y- int(h\_wm/2) left\_x = center\_x- int(w\_wm/2)

bottom\_y = top\_y + h\_wmright\_x = left\_x + w\_wm

roi = img[top\_y:bottom\_y, left\_x:right\_x]

result = cv2.addWeighted(roi, 1, wm, 0.3, 0)

img[top\_y:bottom\_y, left\_x:right\_x] = result cv2.imshow("Watermarked Image", img) cv2.waitKey(0)

cv2.destroyAllWindows()

27)

import cv2

import numpy as np

image = cv2.imread("C:/Users/divya/OneDrive/Pictures/Saved Pictures/cat.jfif")

img2 = cv2.imread('C:/Users/divya/OneDrive/Pictures/Saved Pictures/logo.jfif')

print(image.shape)

cv2.imshow("original", image) imageCopy = image.copy()

cv2.circle(imageCopy, (100, 100), 30, (255, 0, 0),-1)

cv2.imshow('image', image)

cv2.imshow('image copy', imageCopy) cropped\_image = image[80:280, 150:330]

cv2.imshow("cropped", cropped\_image)

cv2.imwrite("Cropped Image.jpg", cropped\_image)

dst = cv2.addWeighted(image, 0.5, img2, 0.7, 0)

img\_arr = np.hstack((image, img2))

cv2.imshow('Input Images',img\_arr)

cv2.imshow('Blended Image',dst)

cv2.waitKey(0)

28)

import cv2

import numpy as np

img = cv2.imread("C:/Users/divya/OneDrive/Documents/COMPUTER VISION/Girl with a Cat.png", cv2.IMREAD\_GRAYSCALE)

dx = cv2.Sobel(img, cv2.CV\_64F, 1, 0)

dy = cv2.Sobel(img, cv2.CV\_64F, 0, 1)

edges = cv2.magnitude(dx, dy)

thresh = 100

edges[edges < thresh] = 0

edges[edges >= thresh] = 25

5 cv2.imshow("Edges", edges) cv2.waitKey(0)

cv2.destroyAllWindows()

29)

import numpy as np

img = cv2.imread("C:/Users/divya/Downloads/Girl with a Cat.png", cv2.IMREAD\_GRAYSCALE) kernel = np.ones((5,5), np.uint8)

erosion = cv2.erode(img, kernel, iterations=1)

cv2.imshow("Original", img)

cv2.imshow("Erosion", erosion)

cv2.waitKey(0)

cv2.destroyAllWindows()

30)

import cv2

import numpy as np

img = cv2.imread("C:/Users/divya/Downloads/Girl with a Cat.png", cv2.IMREAD\_GRAYSCALE)

kernel = np.ones((5,5), np.uint8)

dilation = cv2.dilate(img, kernel, iterations=1)

cv2.imshow("Original", img)

cv2.imshow("Dilation", dilation)

cv2.waitKey(0)

cv2.destroyAllWindows()

31)

import cv2

import numpy as np

img = cv2.imread("C:/Users/divya/Downloads/Girl with a Cat.png", cv2.IMREAD\_GRAYSCALE) kernel = np.ones((5,5), np.uint8)

opening = cv2.morphologyEx(img, cv2.MORPH\_OPEN, kernel)

cv2.imshow("Original", img)

cv2.imshow("opening", opening) cv2.waitKey(0)

cv2.destroyAllWindows()

32)

import cv2

import numpy as np

img = cv2.imread("C:/Users/divya/Downloads/Girl with a Cat.png", cv2.IMREAD\_GRAYSCALE) kernel = np.ones((5,5), np.uint8)

closing = cv2.morphologyEx(img, cv2.MORPH\_CLOSE, kernel)

cv2.imshow("Original", img)

cv2.imshow("Closing", closing)

cv2.waitKey(0)

cv2.destroyAllWindows()

33)

import cv2

import numpy as np

img = cv2.imread("C:/Users/divya/OneDrive/Documents/COMPUTER VISION/Girl with a Cat.png", cv2.IMREAD\_GRAYSCALE)

kernel = np.ones((5,5), np.uint8)

grad = cv2.morphologyEx(img, cv2.MORPH\_GRADIENT, kernel) cv2.imshow("Original", img)

cv2.imshow(“Gradient", grad) cv2.waitKey

34)

import cv2

import numpy as np

img = cv2.imread("C:/Users/koppo/Downloads/Genshin-Impact\_Key-Art-EN-920x518.png", cv2.IMREAD\_GRAYSCALE)

kernel = np.ones((5,5), np.uint8)

tophat = cv2.morphologyEx(img, cv2.MORPH\_TOPHAT, kernel)

cv2.imshow("Original", img)

cv2.imshow("Top Hat", tophat)

cv2.waitKey(0)

cv2.destroyAllWindows()

35)

import cv2

import numpy as np

img = cv2.imread("C:/Users/divya/OneDrive/Documents/COMPUTER VISION/Girl with a Cat.png", cv2.IMREAD\_GRAYSCALE)

kernel = np.ones((5,5), np.uint8)

blackhat = cv2.morphologyEx(img, cv2.MORPH\_BLACKHAT, kernel) cv2.imshow("Original", img)

cv2.imshow("Black Hat", blackhat)

cv2.waitKey(0)

cv2.destroyAllWindows()