CODE

Feature extraction

# Importing the libraries

import cv2

import numpy as np

#Haris Corner Detection

# Reading the image and converting the image to B/W

img = cv2.imread('C:/Users/student/Downloads/images (1).jfif')

image = cv2.imread('C:/Users/student/Downloads/images (1).jfif')

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

gray\_image = np.float32(gray\_image)

# Applying the function

dst = cv2.cornerHarris(gray\_image, blockSize=2, ksize=3, k=0.04)

# dilate to mark the corners

dst = cv2.dilate(dst, None)

image[dst > 0.01 \* dst.max()] = [0,0,255]

Hori = np.concatenate((img, image), axis=1)

cv2.imshow('haris\_corner', Hori)

cv2.waitKey()

#SIFT Detection

# Reading the image and converting into B/W

image = cv2.imread('C:/Users/student/Downloads/images (1).jfif')

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Applying the function

sift = cv2.SIFT\_create()

kp, des = sift.detectAndCompute(gray\_image, None)

# Applying the function

kp\_image = cv2.drawKeypoints(image, kp, None, color=(

      0,0,255), flags=cv2.DRAW\_MATCHES\_FLAGS\_DRAW\_RICH\_KEYPOINTS)

Hori = np.concatenate((img, kp\_image), axis=1)

cv2.imshow('SIFT', Hori)

cv2.waitKey()

#FAST Corner Detection

# Reading the image and converting into B/W

image = cv2.imread('C:/Users/student/Downloads/images (1).jfif')

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Applying the function

fast = cv2.FastFeatureDetector\_create()

fast.setNonmaxSuppression(False)

# Drawing the keypoints

kp = fast.detect(gray\_image, None)

kp\_image = cv2.drawKeypoints(image, kp, None, color=(0, 0,255))

Hori = np.concatenate((img, kp\_image), axis=1)

cv2.imshow('FAST', Hori)

cv2.waitKey()

cv2.destroyAllWindows()

OUTPUT

|  |  |  |
| --- | --- | --- |
| ORIGINAL IMAGE  C:\Users\student\Downloads\images (1).jfif | | |
| HARIS CORNER EXTRACTION | SIFT EXTRACTION | FAST EXTRACTION |