**Task 3**: Masking

**Code:**

**1. Black Blackground mask-**

import cv2

import numpy as np

img1 = cv2.imread("matplot.png", 1)

img2 = cv2.imread("flower.png", 1)

rows, cols, ch = img2.shape

roi = img1[0:rows, 0:cols]

img2gray = cv2.cvtColor(img2, cv2.COLOR\_BGR2GRAY)

ret, mask = cv2.threshold(img2gray, 35, 255, cv2.THRESH\_BINARY)

mask\_inv = cv2.bitwise\_not(mask)

img1\_bg = cv2.bitwise\_and(roi, roi, mask = mask\_inv)

img2\_fg = cv2.bitwise\_and(img2, img2, mask = mask)

dst = cv2.add(img1\_bg, img2\_fg)

img1[0:rows, 0:cols] = dst

cv2.imshow('img', img1)

cv2.imshow('img2', img2)

cv2.waitKey(0)

cv2.destroyAllWindows()

**2.** **HSV Trackbar and Colour detection-**

import cv2

import numpy as np

def nothing(x):

return 0

cap = cv2.VideoCapture(0)

cv2.namedWindow('HSV')

cv2.createTrackbar('Hue Low', 'HSV', 0, 180, nothing)

cv2.createTrackbar('Hue High', 'HSV', 0, 180, nothing)

cv2.createTrackbar('Saturation Low', 'HSV', 0, 255, nothing)

cv2.createTrackbar('Saturation High', 'HSV', 0, 255, nothing)

cv2.createTrackbar('Value Low', 'HSV', 0, 255, nothing)

cv2.createTrackbar('Value High', 'HSV', 0, 255, nothing)

while True:

\_, frame = cap.read()

cv2.imshow('frame', frame)

hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

hl = cv2.getTrackbarPos('Hue Low', 'HSV')

sl = cv2.getTrackbarPos('Saturation Low', 'HSV')

vl = cv2.getTrackbarPos('Value Low', 'HSV')

hh = cv2.getTrackbarPos('Hue High', 'HSV')

sh = cv2.getTrackbarPos('Saturation High', 'HSV')

vh = cv2.getTrackbarPos('Value High', 'HSV')

lower = np.array([hl, sl, vl])

upper = np.array([hh, vh, sh])

mask = cv2.inRange(hsv, lower, upper)

des = cv2.bitwise\_and(frame, frame, mask=mask)

cv2.imshow('HSV', des)

if cv2.waitKey(1) & 0xFF == ord('q'): break

cap.release()

cv2.destroyAllWindows()

**Problems:**

I was facing issues implementing the HSV values using a trackbar.

I tried converting the BGR image to HSV and then directly applying the values to the image as is done in BGR, which ofcourse didn’t work.

**Solutions:**

I looked into the previous HSV program codes and figured out the exact procedure to implement the HSV values properly to the image.