Exp 10:

Object detection

**Code:**

import numpy as np

import cv2 as cv

# Read the input video

cap = cv.VideoCapture('objdetect.mp4')

# Take the first frame of the video

ret, frame = cap.read()

# Setup initial region of tracker

x, y, width, height = 400, 440, 150, 150

track\_window = (x, y, width, height)

# Set up the Region of Interest for tracking

roi = frame[y:y + height, x:x + width]

# Convert ROI from BGR to HSV format

hsv\_roi = cv.cvtColor(roi, cv.COLOR\_BGR2HSV)

# Perform masking operation

mask = cv.inRange(hsv\_roi, np.array((0., 60., 32.)), np.array((180., 255., 255)))

roi\_hist = cv.calcHist([hsv\_roi], [0], mask, [180], [0, 180])

cv.normalize(roi\_hist, roi\_hist, 0, 255, cv.NORM\_MINMAX)

# Setup the termination criteria, either 15 iterations or move by at least 2 pixels

term\_crit = (cv.TERM\_CRITERIA\_EPS | cv.TERM\_CRITERIA\_COUNT, 15, 2)

while True:

ret, frame = cap.read()

# Resize the video frames

frame = cv.resize(frame, (720, 720), fx=0, fy=0, interpolation=cv.INTER\_CUBIC)

cv.imshow('Original', frame)

# Perform thresholding on the video frames

ret1, frame1 = cv.threshold(frame, 180, 155, cv.THRESH\_TOZERO\_INV)

# Convert from BGR to HSV format

hsv = cv.cvtColor(frame1, cv.COLOR\_BGR2HSV)

dst = cv.calcBackProject([hsv], [0], roi\_hist, [0, 180], 1)

# Apply Camshift to get the new location

ret2, track\_window = cv.CamShift(dst, track\_window, term\_crit)

# Draw it on the image

pts = cv.boxPoints(ret2)

pts = np.int0(pts)

# Draw tracking window on the video frame

result = cv.polylines(frame, [pts], isClosed=True, color=(0, 255, 255), thickness=2)

cv.imshow('Camshift', result)

# Set ESC key as the exit button

k = cv.waitKey(30) & 0xff

if k == 27:

break

# Release the cap object

cap.release()

# Close all opened windows

cv.destroyAllWindows()

Output:

**DOWNLOAD A SHORT VIDEO WHERE A LIVING/NON LIVING BEING IN MOVEMENT CONTINUOUSLY**