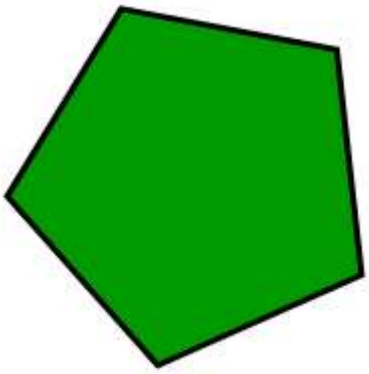
Project work------- Computer Vision module application for finding a target in a live camera



, it’s time to set up a test field. Get several printouts of the target and paste it at several places in your house. Now, if we really want to get the feel, make a quadcopter, fix a small camera in it, and record the whole house properly and ensure that we cover the places where we have pasted the targets. In case if we don’t want to go through all this trouble, just grab a camera and record our house yourself.

IMPLEMENTED CODE

```
import argparse

import cv2


# construct the argument parse and parse the arguments

ap = argparse.ArgumentParser()

ap.add_argument("-v", "--video", help="path to the video file")

args = vars(ap.parse_args())


# load the video

camVideo = cv2.VideoCapture(args["video"])


# keep looping

while True:


    # grab the current frame and initialize the status text
    (grabbed, frame) = camVideo.read()
    status = "No Target in sight"

    # check to see if we have reached the end of the
    # video
    if not grabbed:
        break

    # convert the frame to grayscale, blur it, and detect edges
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #grayscale
    blurred = cv2.GaussianBlur(gray, (7, 7), 0) #blur
    edged = cv2.Canny(blurred, 50, 150) #canny edge detection

    # find contours in the edge map
    (cnts, _) = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
# loop over the contours
for cnt in cnts:
    approx=cv2.approxPolyDP(cnt,0.01*cv2.arcLength(cnt,True),
    True)

if len(approx)==5:
    cv2.drawContours(frame, [approx], -1, (0, 0, 255), 4)
    status = "Target(s) in sight!"

    # draw the status text on the frame
    cv2.putText(frame, status, (20, 30), cv2.FONT_HERSHEY_SIMPLEX,
    0.5,(0, 0, 255), 2)

    # show the frame and record if a key is pressed
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    # if the 's' key is pressed, stop the loop
    if key == ord("s"):
        break

    # cleanup the input recorded video and close any open windows

camVideo.release()
cv2.destroyAllWindows(
```

EXPLANATION:

We loop over each frame of the recorded video and for detection of our target we convert it to gray-scale, blur it, and finally use canny edge detection method to find the outlined image.

Remember that, camVideo.read() will return a tuple with first element specifying whether the frame was read successfully or not, second element is the actual frame we will be working on!

Now, once we have the frame, we will use contour approximation and then check the number of elements in the output obtained from previous step. If the number of elements is 5, then we have the regular pentagon that we were looking for, and hence we update the status.