```
import numpy as np
import cv2


cap = cv2.VideoCapture(0)
while True:
    ret,img=cap.read()
    #Here we store image. Confirm that you save the images in the same directory of this sol4.py file

    rng = cv2.selectROI(img)
    #Here we select the range for shape dedection

    imcrop = img[int(rng[1]) : int(rng[1] + rng[3]), int(rng[0]) : int(rng[0] + rng[2])]
    #Here we take image from selected range

    thresh = 20 # we approx the threshold value

    bmin = imcrop[:, :, 0].min()
    gmin = imcrop[:, :, 1].min()
    rmin = imcrop[:, :, 2].min()
    #Lower bound for threhold

    bmax = imcrop[:, :, 0].max()
    gmax = imcrop[:, :, 1].max()
    rmax = imcrop[:, :, 2].max()
    #Upper bound for threshold

    bgrmin = np.array([bmin - thresh, gmin - thresh, rmin - thresh])
    bgrmax = np.array([bmax + thresh, gmax + thresh, rmax + thresh])
    #Minus and plus are just for good thrsholding, thresh == 20 is not fix we change as we need

    threshold = cv2.inRange(img, bgrmin, bgrmax)
    """
     before find contours we can done some marphological operation like
     cv2.morphologyEx(mask2, cv2.MORPH_OPEN, np.ones((5,5),np.uint8)) and
     dilate also for drawing good contours and ignoring noise
    """

    _, contours, _ = cv2.findContours(threshold, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    #Here we store contours of thresh image. In some opencv verson cv2.findContours return only two
    values COntours, _  .

    for cnt in contours :
        area = cv2.contourArea(cnt)
        if area > 100 : # this is done for ingonering the noise you also done something else like.
            approx = cv2.approxPolyDP(cnt, 0.04*cv2.arcLength(cnt, True), True)

            cen = cv2.moments(cnt) #  we find the centre of that shape we can approx ravel.
            cenx = int(cen["m10"]/cen["m00"])
            ceny = int(cen["m01"]/cen["m00"])

            if(len(approx) == 3) :
```

```
            cv2.putText(img, "Triangle", (cenx, ceny), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255,255,255))

        elif(len(approx) == 4) :
            cv2.putText(img, "Square", (cenx, ceny), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255,255,255))
            #we also classified between square and rectange by using aspect ratio

        elif(len(approx) == 5) :
            cv2.putText(img, "Pentagon", (cenx, ceny), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255,255,255))

        elif(len(approx) == 6) :
            cv2.putText(img, "Hexagon", (cenx, ceny), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255,255,255))

        else :
            cv2.putText(img, "Circle", (cenx, ceny), cv2.FONT_HERSHEY_COMPLEX, 0.5, (255,255,255))
            #we don't know the point for circle so we put in last so if the shape doest not fall in any if then it circle

cv2.imshow("change", img)

"""
  you can don all thing different color space like hsv, grayscal,
  but I do in rgb
"""
```