```
import cv2

cap = cv2.VideoCapture(0)
cap.set(3, 640) # width
cap.set(4, 480) #height

min_area = 550
count = 0

while True:
    # Capture video frame
    success, img = cap.read()
    # algorithm used haarcascade_russian_plate_number
    plate_cascade = cv2.CascadeClassifier(cv2.data.haarcascades
+"haarcascade_russian_plate_number.xml")
    # Convert color to grayscale
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # 1.3=scale factor 2= minNeighbors
    plates = plate_cascade.detectMultiScale(img_gray, scaleFactor=2,
minNeighbors=2)

    # Loops for each plates
    for (x,y,w,h) in plates:
        area = w * h

        if area > min_area:
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
# BGR  2 show thickness
            cv2.putText(img, "Number Plate", (x,y-5),
cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255, 0, 255), 2)
            # img_roi = img[y: y+h, x:x+w]
            # cv2.imshow("ROI", img_roi)
            # cv2.imwrite("plates/scaned_img_" + str(count) + ".jpg",
img_roi)
            cv2.imwrite(f"dataset/scaned_img_" + str(count) + ".jpg",
img_gray[y:y + h, x:x + w])
            count += 1


    cv2.imshow("Frame", img)
    k = cv2.waitKey(1)
    # press enter to exit
    if k == 13 :
        break
    # elif count>=5:
    #     break

cap.release()
cv2.destroyAllWindows()
```