In [1]:

```
import cv2
import pytesseract
import numpy as np
from PIL import ImageGrab
import time
import re
import matplotlib.pyplot as plt
import re, string
pattern = re.compile('[\W_]+')
import statistics
from statistics import mode

pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'
```

In [17]:

```
def detect_plate3(image):
    copy = image.copy()
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    edged = cv2.Canny(gray,100, 300)
    rect_kern = cv2.getStructuringElement(cv2.MORPH_RECT, (2,2))
    # apply dilation to make regions more clear
    dilation = cv2.dilate(edged, rect_kern, iterations = 1)
    contours, new  = cv2.findContours(dilation, cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
    contours=sorted(contours, key = cv2.contourArea, reverse = True)[:10]
    text_list=[]
    for contour in contours:
        perimeter = cv2.arcLength(contour, True)
        approx = cv2.approxPolyDP(contour, 0.07* perimeter, True)
        #cv2.drawContours(image2,contour,-1,(0,0,255),3)
        if len(approx)>=4 and cv2.contourArea(approx) > 3000 and cv2.contourArea(approx) < 5000:
            x, y, w, h = cv2.boundingRect(approx)
            #print(cv2.contourArea(contour))
            #print(len(approx))
            #cv2.rectangle(image2, (x , y ), (x + w , y + h ), (0, 255, 0), 3)
            license_plate = gray[y:y + h, x:x + w]
            b_filter = cv2.bilateralFilter(license_plate, 11, 17, 17)
            #(thresh, license_plate_threshed) = cv2.threshold(b_filter, 150, 180, cv2.THRESH_BINARY)
            (thresh, license_plate_threshed) = cv2.threshold(b_filter, 0, 255, cv2.THRESH_OTSU | cv2.TH
RESH_BINARY)
            text = pytesseract.image_to_string(license_plate_threshed, config='-c tessedit_char_whiteli
st=0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ --psm 6')
            clean_text = pattern.sub('', text)
            if clean_text != '' and len(clean_text)==6:
                #text_list.append(clean_text)
                cv2.rectangle(copy, (x , y ), (x + w , y + h ), (0, 255, 0), 3)
                cv2.putText(copy, clean_text,(x-100,y-50), cv2.FONT_HERSHEY_SIMPLEX, 2, (255, 0, 0), 2,
 cv2.LINE_AA)
                #print(clean_text)
                #print(len(approx))
                #print(cv2.contourArea(approx))
                print(clean_text)
                return copy
```

In [18]:

```
#Initilize Video capture Object and points towards a video file
cap = cv2.VideoCapture('video3.mp4')
fourcc = cv2.VideoWriter_fourcc(*'MP4V')
out = cv2.VideoWriter(r'detectvid.mp4',fourcc, 20.0, (int(cap.get(3)), int(cap.get(4))))


# Lets print the status of your Capture.
print('Status:',cap.isOpened())

# Initilize a loop in which we will read video frame by frame
while(True):
    ret, frame = cap.read()

# if a frame is not read correctly or there are no more frames to be read the
    if not ret:
        break

    output = detect_plate3(frame)
    out.write(output)

out.release()
cap.release()
```

```
Status: True
BND500
BND500
BND500
BND500
BND500
BNO500
BND500
BND500
BND500
BND500
BND500
BND500
BND500
BND500
BNO500
BND500
BND500
BND500
BNO500
```

In [ ]: