

LAMPIRAN

```

import cv2
import pytesseract

pytesseract.pytesseract.tesseract_cmd = 'C:/Program Files (x86)/Tesseract-OCR/tesseract'

# Read the image file
image = cv2.imread('car2.JPG')
cv2.imshow("Original",image)
# Convert to Grayscale Image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

#Canny Edge Detection
canny_edge = cv2.Canny(gray_image, 170, 200)

# Find contours based on Edges
contours, new = cv2.findContours(canny_edge.copy(), cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
contours=sorted(contours, key = cv2.contourArea, reverse = True)[:30]

# Initialize license Plate contour and x,y,w,h coordinates
contour_with_license_plate = None
license_plate = None
x = None
y = None
w = None
h = None

# Find the contour with 4 potential corners and create ROI around it
for contour in contours:
    # Find Perimeter of contour and it should be a closed contour
    perimeter = cv2.arcLength(contour, True)
    approx = cv2.approxPolyDP(contour, 0.01 * perimeter, True)

```

```

if len(approx) == 4:                                #see whether it is a Rect
    contour_with_license_plate = approx
    x, y, w, h = cv2.boundingRect(contour)
    license_plate = gray_image[y:y + h, x:x + w]
    break
(thresh, license_plate) = cv2.threshold(license_plate, 127, 255,
cv2.THRESH_BINARY)
cv2.imshow("plate",license_plate)
# Removing Noise from the detected image, before sending to Tesseract
license_plate = cv2.bilateralFilter(license_plate, 11, 17, 17)
(thresh, license_plate) = cv2.threshold(license_plate, 150, 180,
cv2.THRESH_BINARY)

#Text Recognition
text = pytesseract.image_to_string(license_plate)
#Draw License Plate and write the Text
image = cv2.rectangle(image, (x,y), (x+w,y+h), (0,0,255), 3)
image = cv2.putText(image, text, (x-100,y-20),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2, cv2.LINE_AA)

print("License Plate :", text)

cv2.imshow("License Plate Detection",image)
cv2.waitKey(0)

```