In [*]:

```python
from imutils.object_detection import non_max_suppression
import numpy as np
import time
import cv2
import pytesseract

net = cv2.dnn.readNet("frozen_east_text_detection.pb")


def text_detector(image):
    orig = image
    (H, W) = image.shape[:2]

    (newW, newH) = (320, 320)
    rW = W / float(newW)
    rH = H / float(newH)

    image = cv2.resize(image, (newW, newH))
    (H, W) = image.shape[:2]

    layerNames = [
        "feature_fusion/Conv_7/Sigmoid",
        "feature_fusion/concat_3"]


    blob = cv2.dnn.blobFromImage(image, 1.0, (W, H),
        (123.68, 116.78, 103.94), swapRB=True, crop=False)

    net.setInput(blob)
    (scores, geometry) = net.forward(layerNames)

    (numRows, numCols) = scores.shape[2:4]
    rects = []
    confidences = []

    for y in range(0, numRows):

        scoresData = scores[0, 0, y]
        xData0 = geometry[0, 0, y]
        xData1 = geometry[0, 1, y]
        xData2 = geometry[0, 2, y]
        xData3 = geometry[0, 3, y]
        anglesData = geometry[0, 4, y]

        for x in range(0, numCols):

            if scoresData[x] < 0.5:
                continue
            (offsetX, offsetY) = (x * 4.0, y * 4.0)
            angle = anglesData[x]
            cos = np.cos(angle)
            sin = np.sin(angle)
            h = xData0[x] + xData2[x]
            w = xData1[x] + xData3[x]
            endX = int(offsetX + (cos * xData1[x]) + (sin * xData2[x]))
            endY = int(offsetY - (sin * xData1[x]) + (cos * xData2[x]))
            startX = int(endX - w)
            startY = int(endY - h)
            rects.append((startX, startY, endX, endY))
            confidences.append(scoresData[x])

    boxes = non_max_suppression(np.array(rects), probs=confidences)

    for (startX, startY, endX, endY) in boxes:

        startX = int(startX * rW)
        startY = int(startY * rH)
        endX = int(endX * rW)
        endY = int(endY * rH)
        boundary = 12

        text = orig[startY-boundary:endY+boundary, startX - boundary:endX + bound
        text = cv2.cvtColor(text.astype(np.uint8), cv2.COLOR_BGR2GRAY)
        pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OCR
        textRecongized = pytesseract.image_to_string(text)
        cv2.rectangle(orig, (startX, startY), (endX, endY), (0, 255, 0), 3)
        orig = cv2.putText(orig, textRecongized, (endX,endY+5), cv2.FONT_HERSHEY_
    return orig

image0 = cv2.imread('image1.jpg')

array = [image0]

for i in range(0,2):
    for img in array:
        car_wash = cv2.resize(img, (640,320), )
        orig = cv2.resize(img, (640,320), )
        textDetected = text_detector(image0)
        cv2.imshow("Original Image",orig)
        cv2.imshow("Text Detection", textDetected)
        time.sleep(2)
        k = cv2.waitKey(300000)
        if k == 27:
            break
cv2.destroyAllWindows()
```