**IMPORTING PACKAGES AND DISPLAYING IMAGE:**

1. Importing necessary packages

Opencv,Numpy,pytesseract

2. Path of pytesseract is mentioned using the pytesseract command.

3. Reading the image using opencv function imread () and displaying the same using imshow ().

4. Converting the image into greyscale using cv2.cvtcolor () by passing the image andcv2.COLOR\_BGR2GRAY.





****

**LICENSE PLATE DETECTION:**

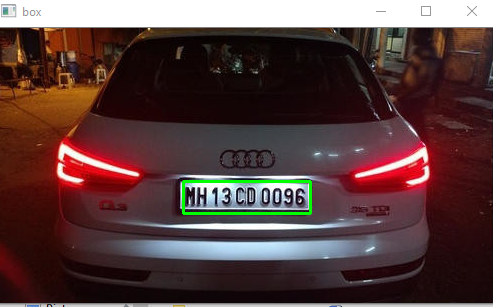
5. To localize the license plate, a machine learning algorithm named haarcascade which is pre-trained with several “positive” sample views of a particular object and arbitrary “negative” images of the same size is used.

6. Using cascadeClassifier () function, we apply the algorithm to the region of the image and in turn it detects the object in question.

7. With detectMultiScale the detected objects are returned as a list of rectangles.

(Reference for Cascade classifier: <https://sites.google.com/site/5kk73gpu2012/assignment/viola-jones-face-detection#TOC-Image-Pyramid>)

8. Using for loop, bounding box is made for the license plate and the image is shown as well.


**CHARACTER SEGMENTATION & RECOGNITION:**

9. The image is cropped and it is converted into greyscale.

10. Using threshold function we’re able to classify the pixel intensities in the greyscale image and create binary images.

11.The text is extracted from the license plate using pytesseract function image  
\_to\_string. where config is pgm(page segmentation mode)-11 which indicates that tesseract should only look for one line of text.

(Reference: https://www.tutorialspoint.com/opencv/opencv\_simple\_threshold.html)

14. To place the boxes on characters of license plate I used the image\_to\_boxes function and stored it in the variable char\_box.

**LIST CREATION:**

15. declaring a function Convert (string).

16. An empty list is created for the insertion of license plate characters.

17. The string value gets stored in the list and it is returned and stored in variable named letters.

**DETECTING CHARACTERS AND FEATURES:**

18. Declaring two variables previous and count and initializing them to 0.

19. To detect characters am using for loop so that it loops over char\_box.

20. Height and width of the cropped image is found out using shape () function

21. Using split () function, the characters are stored.

22. Using if condition we have to check if 0th index of b is equal to letters [cnt].

23. Since everything is string I converted x, y, w, h into integer and rectangle is drawn using cv2. Rectangle ().

24. Height, width and space is calculated and displayed. Count variable gets incremented by 1.

25. Previous gets equal to width.

26. Else continue

27. Finally detected license number is shown.



['T', '9', '3', '27', '39', '0']

T: - Height: 36 Width: 18 Space: 9

['N', '31', '4', '51', '48', '0']

N: - Height: 44 Width: 20 Space: 4

['0', '62', '6', '81', '60', '0']

0:- Height: 54 Width: 19 Space: 11

['2', '82', '7', '103', '50', '0']

2:- Height: 43 Width: 21 Space: 1

['B', '112', '8', '132', '44', '0']

B: - Height: 36 Width: 20 Space: 9

['K', '134', '10', '152', '45', '0']

K: - Height: 35 Width: 18 Space: 2

['6', '163', '10', '181', '45', '0']

6:- Height: 35 Width: 18 Space: 11

['7', '188', '12', '203', '60', '0']

7:- Height: 48 Width: 15 Space: 7

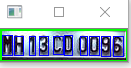
['5', '210', '15', '228', '49', '0']

5:- Height: 34 Width: 18 Space: 7

['1', '233', '15', '243', '50', '0']

1:- Height: 35 Width: 10 Space: 5

License plate dimensions: 62 x 248



['M', '2', '6', '12', '26', '0']

M: - Height: 20 Width: 10 Space: 2

['H', '14', '6', '23', '25', '0']

H: - Height: 19 Width: 9 Space: 2

['1', '29', '6', '35', '25', '0']

1:- Height: 19 Width: 6 Space: 6

['3', '39', '6', '48', '25', '0']

3:- Height: 19 Width: 9 Space: 4

['C', '53', '7', '63', '25', '0']

C: - Height: 18 Width: 10 Space: 5

['D', '64', '6', '72', '26', '0']

D: - Height: 20 Width: 8 Space: 1

['0', '80', '6', '87', '25', '0']

0:- Height: 19 Width: 7 Space: 8

['0', '89', '6', '99', '25', '0']

0:- Height: 19 Width: 10 Space: 2

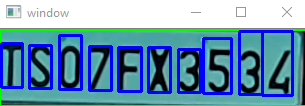
['9', '101', '5', '110', '25', '0']

9:- Height: 20 Width: 9 Space: 2

['6', '113', '5', '123', '25', '0']

6:- Height: 20 Width: 10 Space: 3

License plate dimensions: 32 x 127



['T', '2', '19', '23', '63', '0']

T: - Height: 44 Width: 21 Space: 2

['S', '30', '18', '52', '61', '0']

S: - Height: 43 Width: 22 Space: 7

['O', '60', '17', '82', '71', '0']

O: - Height: 54 Width: 22 Space: 8

['7', '90', '16', '112', '60', '0']

7:- Height: 44 Width: 22 Space: 8

['F', '119', '15', '142', '59', '0']

F: - Height: 44 Width: 23 Space: 7

['X', '149', '14', '171', '59', '0']

X: - Height: 45 Width: 22 Space: 7

['3', '179', '14', '201', '57', '0']

3:- Height: 43 Width: 22 Space: 8

['5', '204', '13', '232', '68', '0']

5:- Height: 55 Width: 28 Space: 3

['3', '240', '12', '263', '75', '0']

3:- Height: 63 Width: 23 Space: 8

['4', '263', '11', '293', '75', '0']

4:- Height: 64 Width: 30 Space: 0

['|', '293', '1', '309', '75', '0']

License plate dimensions: 77 x 310



['M', '21', '14', '37', '45', '0']

M :- Height: 31 Width: 16 Space: 21

['H', '42', '15', '59', '46', '0']

H :- Height: 31 Width: 17 Space: 5

['O', '67', '16', '83', '47', '0']

O :- Height: 31 Width: 16 Space: 8

['Z', '89', '17', '105', '48', '0']

Z :- Height: 31 Width: 16 Space: 6

['E', '113', '18', '130', '50', '0']

E :- Height: 32 Width: 17 Space: 8

['P', '133', '19', '150', '65', '0']

P :- Height: 46 Width: 17 Space: 3

['1', '162', '20', '172', '51', '0']

1 :- Height: 31 Width: 10 Space: 12

['5', '180', '21', '196', '52', '0']

5 :- Height: 31 Width: 16 Space: 8

['4', '202', '23', '218', '54', '0']

4 :- Height: 31 Width: 16 Space: 6

['3', '223', '1', '268', '65', '0']

3 :- Height: 64 Width: 45 Space: 5

License plate dimensions: 67 x 269