1. Importing Libraries:
   1. cv2 to read, view and apply filters on the image.
   2. Pytesseract to read text from the image
   3. Numpy to make matrices and do mathematical operations
   4. Os for dealing with images and paths
   5. Re for regular expression.
2. Defined a function find\_contours with image,mask,lower and upper as paramters:

8.1 Use cv2 find contours on the mask.

8.2 Create a copy of original image.

8.3 Use cv2 draw contours to draw the contours on the image.

8.4 Sort the top 10 contours in descending order.

8.5 Iterate through all the contours:

8.5.1 Get perimeter of the contour using cv2 arc length.

8.5.2 Get the contour with polygonal curve to get the approx. number plate contour.

8.5.3 Check if approx contour has four points:

8.5.3.1 Set approx. as number plate contour.

8.5.3.2 Get x,y,w,h of the contour using bounding rect.

8.5.3.3 Draw a rectangle on these coordinates.

8.5.3.4 Create a crop\_img of number plate using these coordinates.

8.5.4 If not 4 points:

8.5.4.1 Convert image to HSV

8.5.4.2 Create mask according to lower and upper parameters.

8.5.4.3 Dilate the mask to make ROI prominent.

8.5.4.4 Find contours on the dilated image.

8.5.4.5 Create a copy of original image.

8.5.4.6 Draw contours on the original image.

8.5.4.7 Sort the contours in descending order.

8.5.4.8 For Contours of area>500 we will get x,y,w,h.

1. Defined a write function which will image and count as parameters. It will save the image in the directory with name as count and will return count with an increment of 1

1. Main function:

11.1 Define the path of the files.

11.2 Loop through all the files in the folder and read them through opencv.

11.3 Set the upper and lower values of the kernel for mask.

11.4 Convert the image to hsv

11.5 Create the mask of the converted image.

11.6 Call the find\_contours function for number plate processing, If any error occurs call Number Plate Not Found exception.

11.7 Write the processed image in the results folder.