**Easy\_OCR\_approach:**

1. **Firstly input the necessary dependencies.**
2. **Import the image**
3. **Convert into gray scale**
4. **Apply bilateral filter for noise reduction and smoothning of photo**
5. **Apply canny filter on the bilateral filtered image**
6. **Finding keypoints of the image**
7. **Using imutils.grab to extract the number of contours found**
8. **The code checks if the contour has 4 edges, then it's most likely to be a number plate.**
9. **Then we crop the image.**

**Results: This code works fine for some images with better quality. But it needs a change in hyper-parameter for detection. Sometimes it detects wrong objects.**

**Fast RCNN approach:**

1. **Importing the pre-trained fast rcnn model**
2. **Converting the image to tensor and then passing this tensor to the model.**
3. **Then defined a list of categories.**
4. **This results in labels, boxes and their confidence score**
5. **Then filtered the boxes with a confidence level above of 0.8**
6. **In the for loop, extracting the upper left x, y coordinates and lower right x, y coordinates**
7. **Then matching the identity with any member of the pre-defined list.**
8. **Draw bounding boxes.**
9. **Then put text on each bounding box.**

**Results: This code successfully draw boxes around some objects in the image, it also make rectangle around number plate but with new category each time. Till yet, I was not able to solve this particular issue of automatic number plate recognition.**