License Plate Recognition System using Deep Learning

Vivek kumar

Abstract—Number plate recognition is currently a research area for much. Number plate recognition is a technique using which vehicle number is identified. This paper shows the implementation of Deep Learning convolutional neural networks for detection and identification of the images of handwritten characters into editable text format. We make use of convolutional neural networks method for identification of number plate of vehicle and image denoising for get noise free image. Image segmentation is done to overcome the problem of waged, tilted images. The string formed is searched for standard format using Regular Expression. The extracted number can be used with / without database in many ways like electronic payment systems (challan, parking fee payment, toll payment).

Index Terms- Number plate, license plate recognition, Deep Learning, VGG16, convolutional neural network, DnCNN-3, Image Denoising

1 INTRODUCTION

Everyone owns a vehicle these days so it very hard to manage traffic. Significant challenges are faced by countries related to traffic monitoring.one of the way to monitor is number plate.
It is a unique number assigned to a particular vehicle. License plate gives many details about vehicle and the one owning it. The image processing technique -Number plate recognition is to identify the vehicle and its owner by its license plate. The variations of the number plate or the environment at which image is taken can cause difficulty in the recognition and detection of license plates. They can be summed up as:

1) Plate variations:
a) Location: plates can be in any position of image
b) Quantity: it may happen that image have more than one plate
c) Size: sixe may vary due to camera position;
d) Color: plates may have various characters and background colors.
e) Inclination: plates may be tilted;
2) Environment variations:
Input images may have different types of illuminance, mainly due to environmental lighting and vehicle headlights.

The identification of license plate number from a given image is composed of six stages. The First stage is pre-processing where morphological operation along with grey scale conversion is done. The second stage is blob detection/plate extraction to extract the license plate from the image based on features such as boundary. The third stage is segmentation which will overcome all problems faced in previous steps such as tilted plate, non-uniform brightness. Here we extract the characters. The fourth step is character recognition to recognize the extracted characters by template matching. The fifth step is to match the string regular expression. The final stage is to match it with database to get related details.

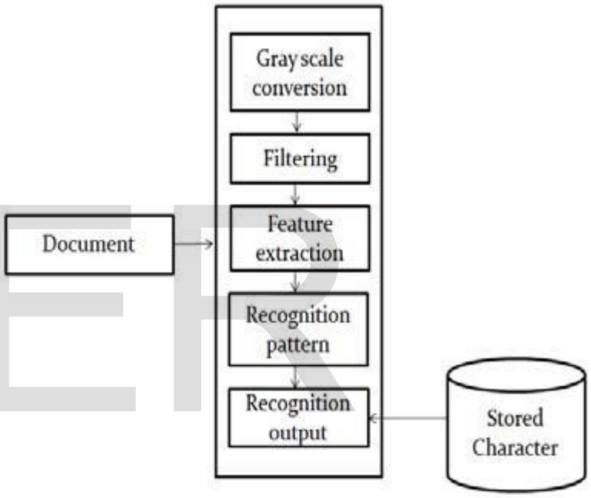

Figure1: System block diagram

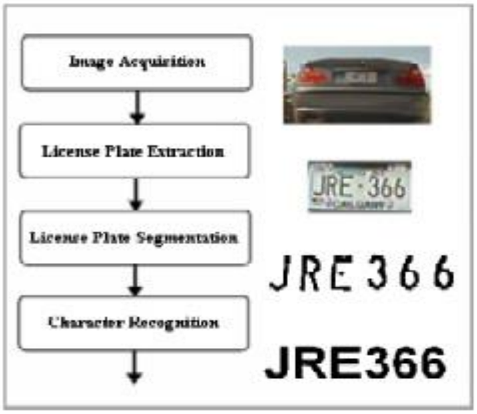

Figure 2: Step wise Processed Images

2 PRE-PROCESSING

In pre-processing the vehicle number plate is converted into grayscale for achieving accuracy. So, input image can be in

- *Vivek Kumar is currently pursuing master's degree program in Computer science – Data Science in Trinity College Dublin, Ireland. PH -+353-899481246. E-mail: Vivekdalmia060@gmail.com* any color as they as converted into grayscale. To convert the RGB color image to grayscale we earlier used the following function.

$$Y = 0.2126R+0.7152G+0.0722B$$

Image denoising is done, over to gray scale image to achieve high quality image from its noisy (degraded) observation. The model for image degradation for denoising task is:

$$yfinal = ximage + n$$

where ximage refers to the unknown high quality image, yfinal is the degraded observation, and n represents the additive noise. The model used here for denoising task is DnCNN-3 which is learns for three image tasks- Gaussian noise, the SISR input generated by first bicubic downsampling and then bicubic upsampling the high-resolution image and the JPEG deblocking input generated by compressing the image with a quality factor ranging from 5 to 99. Combination of all the images was treated as a single input to the DnCNN-3 with the task to remove noise from the image

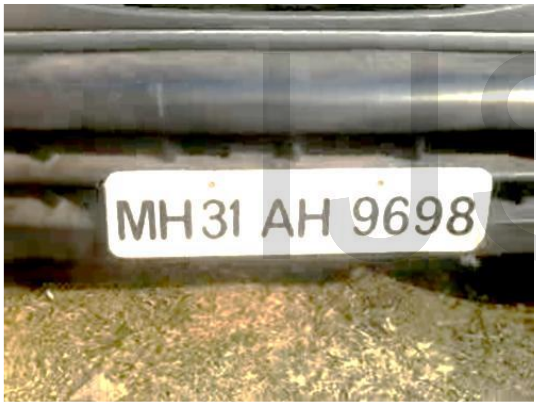

Figure 3: RGB image

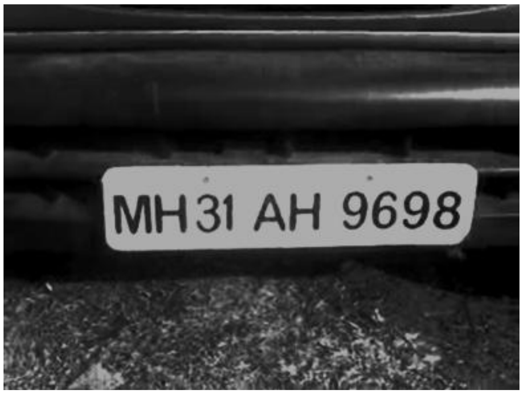

Figure 4: Gray scale frame

3 Plate Extraction

This part focuses on area of image where we can find number/text. The input to this is vehicle image and the output is just the image containing the license number plate.
It is done by using texture feature .Here the gray scaled image is used. The character are identified using the character color and license plate background color.
Few improvements are made in coarse to achieve better predictions- The network uses convolution which extracts the feature from the image and passes it on to the next layer. But while doing so, the spatial information is lost. For doing extraction the spatial information in the earlier layers must be combined with the classification information in the deepest layer. So for FCN-16 the output from pool5 is 2 times upsampled and fuse with pool4 and perform 16 times upsampling. Similarly, for FCN-8 the output from pool5 is 2 times upsampled and fuse with pool4 and perform 8 times upsampling thus improving the coarse prediction by incorporating spatial information from earlier layers

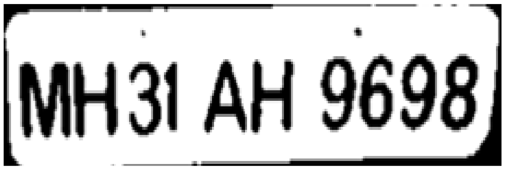

Figure 5: Extracted number plate

4 License plate segmentation

Now, the important part i.e. segmentation for extraction of characters. Some plate may have problems such as wage image, tilted plate, etc. The segmentation algorithm should be used such as to overcome such difficulty like least-squares method can be used to remove horizontal tilt and vertical tilt in the plate.

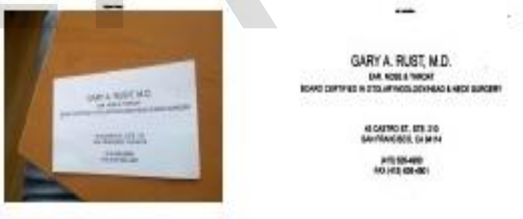

Figure 6: Image Segmentation

The goal of semantic image segmentation is to label each pixel of an image with a corresponding class of what is being represented. Applications for semantic segmentation include: Autonomous driving, Industrial inspection, Classification of terrain visible in satellite imagery, Medical imaging analysis and many more. Segmentation can be done using Pixel Connectivity. Here, we label the connecting pixels to binary form. The labelled pixel are then analyzed for which having the same size and ratio of characters are considered to as license plate characters. Deep Learning models VGG16 - convolutional neural networks are used to train model on various images to extract the text from
FCN (Fully Convolutional Network) is a neural network where multiple convolution layers and are used for semantic segmentation. In convolution, the features are extracted from the image which helps in classification of the image but while doing that the spatial information is lost. To perform semantic segmentation FCN has to give each pixel a probability if it

belongs to a particular class and to do so it uses upsampling. For upsampling the output for the end most layer is upsampled with the prior layer using interpolation techniques, for example, bilinear interpolation technique. While upsampling the size of the output is increased to the original size of the input. This process is repeated for each convolution layer which results in the output having the same size as the input image, but where each pixel is assigned a probability for a particular class.

5 Character Recognition

The extracted characters are now recognized and the output we get is the license plate number. Character recognition may have some difficulties. Because of camera zoom factor, the extracted characters do not get the exact size and the exact thickness. Resizing the characters into one size only before recognition helps to overcome this problem. The font is not the same in all cases since different countries uses different fonts in their license plates. Extracted characters may contain some noise or they may be broken/distort. The extracted characters may also be bend/tilted. In the following, we will categorize the existing character recognition methods based on the features they used.

The Recall and precision are used as metric calculation for sematic segmentation, where recall is defined as correct detection upon corrected plus missed. And precision is defined as correct detection upon corrected plus false positive.

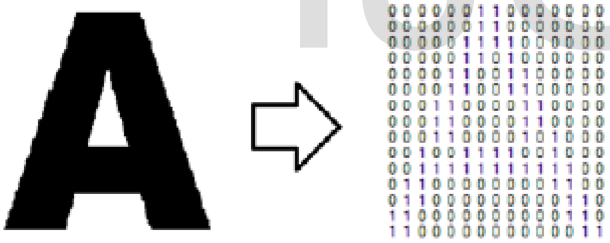

Figure 7: Binarization of Character

6 Extract With Regular Expression (R.E.)

Here, the validity of string is done using string matching technique i.e. Regular expression. The expression is used to get string in particular format. For Indian Number Plate system [i.e. JH 31 AB 3277].

The Regular Expression is:

[A-Z]{2}[0-9]{2}[A-Z]{1,2}[0-9]{3,4}

7 Conclusion

In this paper, discussed the method for extraction, segmentation and finally validation of vehicle number plate. The images once denoised achieve better accuracy in predicting text. The detected text can be matched with database for details such as name of owner, address, date of purchase, etc. Input image can be in RGB. Further work can be done on this for multiple fonts and for different character type and handwritten character.

Its future application can be development of smart web cams, dashboards for cars, etc.

Acknowledgment

The authors would like to thank Prof. Bahman Honari for his advice throughout this project which helped me in successful completion of project.

References

[1] J. Liang, et. al. “Geometric Rectification of Camera-captured Document Images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 591-605, July 2006.

[2] Prof. Bahman Honari, Trinity College Dublin, Email Correspondence.

[3] K. H. Jin, M. T. McCann, E. Froustey and M. Unser, "Deep Convolutional Neural Network for Inverse Problems in Imaging," in *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509-4522, Sept. 2017, doi: 10.1109/TIP.2017.2713099.

[4] Sonia Bhaskar, Nicholas Lavassar, Scott Green “Implementing Optical Character Recognition on the Android Operating System for Business Cards”, EE 368 Digital Image Processing

[5] Faisal Mohammad, Jyoti Anarase, Milan Shingote, Pratik Ghanwat “Optical Character Recognition Implementation”, ISB&M School of Technology, Nande, Pune- 412115Using Pattern Matching”

[6] Rokus Arnold, Poth Miklos,” Character Recognition Using Neutral Networks”, IEEE Computer 978-1-4244-9280-0/10..

[7] S. T. Kahan, T. Pavlidis, and W. Baird,“On recognition of printed characters of any font and size”, IEEE Transactions of Pattern Recognition and Machine Intelligence,PAMI-91987,pp.274-285.