License Plate Number Recognition

Ahmet Yasin Alp, Orbay Altuntoprak, Ozge Bozal

Problem

License plate recognition is a crucial thing for traffic control and for many other legal issues. In our project we formalize our problem as:

1. Detecting license plate in an image
2. Detecting characters of the license plate
3. Character recognition for the plate number

Method

The functional form of the neural network:

$$f(x; W, b) = LogSoftmax(W_2 ReLu(W_1 x + b_1) + b$$

where ReLu is a rectified linear unit which is an activation function defined as the positive part of its argument. W and b represent weight and bias vectors, respectively. Input 'x' is the matrix each row of which represents pixel values of detected characters from the license plate.

For the optimization part 'Adam Algorithm' implemented on Pytorch is used. Adam is different to classical stochastic gradient descent.

Stochastic gradient descent maintains a single learning rate (termed alpha) for all weight updates and the learning rate does not change during training.

A learning rate is maintained for each network weight (parameter) and separately adapted as learning unfolds.

Training

For the optical character recognition part 3 layer neural network is trained using Pytorch[3] on The Chars74K dataset[1], total of 32400 images of characters from computer fonts with 4 variations (combinations of italic, bold and normal) These are digits from 0-9 and letters from A-Z. Our model is tested on 3600 images of the same format.

References

[1] T. E. de Campos, B. R. Babu and M. Varma. Character recognition in natural images In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, February 2009.*

[2] Multimedia Technology Laboratory National Technical University of Athens LPR Database. September,2017 <http://www.medialab.ntua.gr/research/LPRdatabase/Still_images/day/>

[3] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam, Automatic differentiation in PyTorch In *2017.*

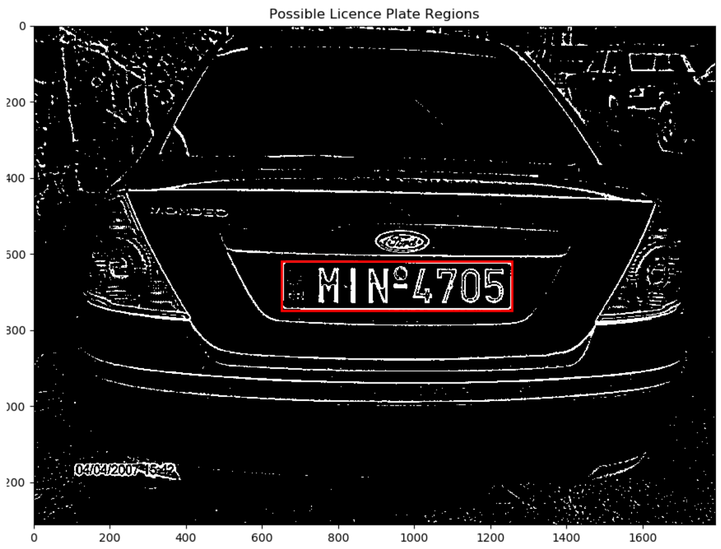
Image Processing

- **Pre-processing the image:**

  First the image is represented with HSV color space, then the value channel is extracted. Since the grayscale conversion of an image is basically a weighted sum of its RGB channels, we chose to use value channel(who is a dedicated dimension to brightness) over grayscale conversion. It empirically proved to be more successful when performing threshold operations to extract character regions. Then the image is blurred using a 5x5 gaussian smoothing kernel. As for the last step, first, Otsu's threshold method is utilized for threshold operations, which was later changed to adaptive treshold operations. This way we obtained less bulky characters and lines. Hence obtained better recognition.

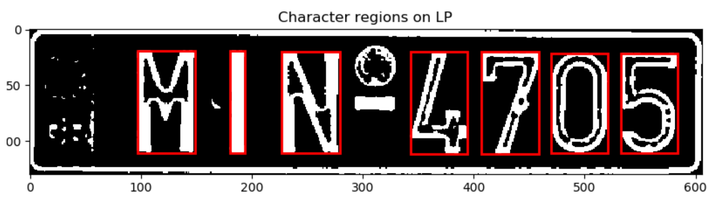
- **Locating the Licence Plate:**

  In order to locate the licence plate in the image, connected component analysis is utilized on the threshold image we obtained at the last step. A 768x1024 image roughly yields a couple thousands of connected regions. This step is all about eliminating those regions to obtain the regions that most likely fall on licence plate area. First, the regions are filtered by their area, the ones that are too small are canceled out. Then the remaining ones are again filtered by their aspect ratios and height & width properties. We let the regions be for this step. It is sometimes the case that we get some false positives, but those are eliminated during the next step.

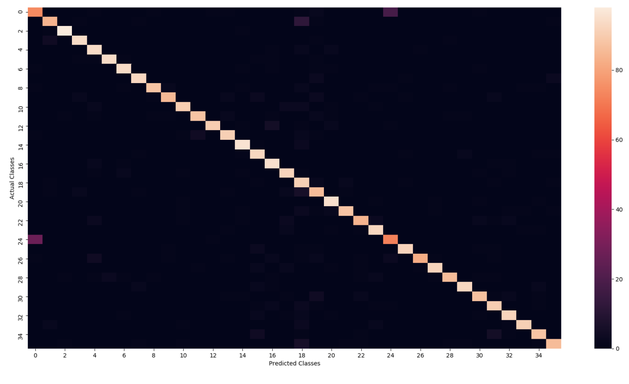


- **Segmentation of Characters:**

  A list of regions are obtained from the previous step. For each element in that list, we do another connected component analysis. We mask the threshold image so that the only 1 region contains non-zero cells at a time. For each masked region, we do a contour analysis pick. Some parameters used to choose between contours are: aspect ratio of the bounding rectangle of the biggest contour we find in regions,



Results



Our model shows highly accurate results on synthetic test data. The confusion matrix of model that we train on synthetic test data is on the left. When we tested on real life car images, the accuracy of finding correct text of license plates is 0.6. The main problem is finding lower number of characters on a given plate. The other one is locating the plate on a given image. These problems mainly stem from image processing part.Future work will be focusing on these issues.