License Plate Recognition in Various Environments with CNN

Yuval Levental
RIT Center for Imaging Science
54 Lomb Memorial Dr, Rochester, NY 14623
yhl3051@rit.edu

Abstract

License plate (LP) recognition, a form of Optical Character Recognition (OCR), is an important modern technology for many legal purposes. It is important to keep track of cars that are parked in various areas, to ensure that there is no illegal parking. It is also useful in tracking cars on the road, for catching individuals that are speeding, commit crimes, or other reasons. One major challenge is creating effective recognition software that can work in many varying street environments. Many LP databases only contain a few thousand license plates in relatively good conditions. However, this project will focus on utilizing convolutional neural networks to recognize many types of license plates in different kinds of configurations and environments. [1][2]

Unfortunately, I was not able to find an effective method to compute accuracy. However, the CCPD dataset did provide a lot of useful information for the training model. Furthermore, the CNN I designed simultaneously performs license plate detection and recognition.

1. Introduction

The main three steps will be to detect the location of the LP with a bounding box, use segmentation to isolate the characters from the background, and to perform accurate character recognition. There are many environmental factors in LP image quality, such as camera angle (XYZ), weather conditions, camera quality, and the time of day. These conditions are common in real world applications.

The dataset that will be used in this project is the Chinese City Parking Dataset (CCPD). It is the largest recorded LP dataset with over 250k images. The filenames of the images contain several coded descriptors such as the LP number, bounding box, LP vertices, tilt angles, among other descriptors. The data set is provided by Plate Recognizer, a company which develops plate recognition software. [3]

The first method that will be used are convolutional neural networks (CNN). CNNs perform a convolution with the input data to detect certain patterns which correspond to certain features. Another method that will be used is pooling, which is used to find the maximum values in sections of an image. Pooling is a form of non-linear downsampling. [4]

Most algorithms assume that the license plate CNN should be separated into two main stages: detection and recognition. One issue with this method is that recognition is dependent on detection, so an imperfect detection could lead to an imperfect recognition. The LP recognition stage could potentially utilize features extracted by the LP detection stage to detect LP characters. Essentially, it can be trained end-to-end.

2. Related Works

2.1. LP Recognition Datasets

Most LP Datasets on Plate Recognizer are not remotely as comprehensive as the CCPD dataset. For instance, the UCSD car dataset only has 1000 images, and the images are from only one parking lot. The UFPR-ALPR Dataset only has 1500 images with a limited license plate color range. Another dataset only has 500 images with a limited range of camera angles. [3]

Additionally, many of these images are either taken from surveillance cameras, or are taken at a close distance at invariant angles. The only requirement for contributors to the CCPD dataset is that the LP number is visible in the picture, which provides many different possible scenarios.

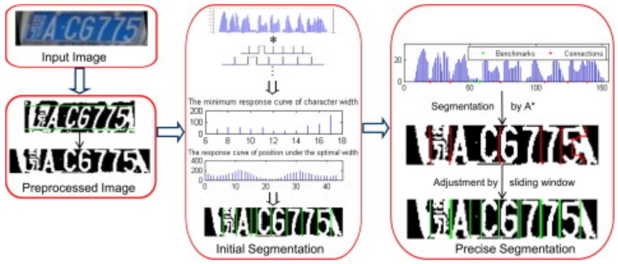
Overall, the CCPD dataset is quite different in the sense that information about distance, tilt degrees, blur, illumination, and weather can be extracted.

2.2. LP detection and recognition algorithms

LP detection originally focused on edge information or background color features. Some of these methods focused on morphology and wave gradients. Other corresponding methods have focused on gradients and colors. Newer methods such as SSD and YOLO eliminate steps such as proposal generation and resampling, using techniques such as network collaboration and mathematical regression.

LP recognition is composed of segmentation-free methods and segmentation-based methods. Segmentation-free methods directly read plate characters, or use a neural network to directly analyze the license plate. Segmentation-based methods use bounding boxes and warping to prepare the image for segmentation. After segmentation, a CNN is used, or other patterns are identified through algorithms like SIFT. [5]

Region proposals have become far more important recently. Faster-RCNN, an example of these networks, utilizes region proposals that detect objects more accurately and quickly. YOLO and its improved version spatially separate bounding boxes and associated class probabilities.[5]

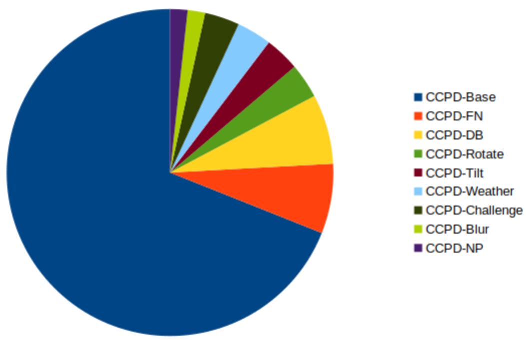


3. CCPD Dataset Introduction

3.1. Image Layout

There are several categories: CCPD-Base, CCPD-FN, CCPD-DB, CCPD-Rotate, CCPD-Tilt, CCPD-Weather, CCPD-Challenge, CCPD-Blur, and CCPD-NP. All these images represent different conditions, which will be explained later.

The main number of images are the Base images, of which there are 200k. CCPD-Blur and CCPD-NP contain only 5000 images each. A pie chart of the image types is provided for reference.



3.2. Image Statistics

	Description
CCPD-Base	The only common feature of these photos is the inclusion of a license plate.
CCPD-DB	Illuminations on the LP area are dark, uneven or extremely bright.
CCPD-FN	The distance from the LP to the shooting location is relatively far or near.
CCPD-Rotate	Great horizontal tilt degree ($20° \sim 50°$) and the vertical tilt degree varies from -10° to 10°.
CCPD-Tilt	Great horizontal tilt degree ($15° \sim 45°$ degrees) and vertical tilt degree ($15° \sim 45°$).
CCPD-Blur	Blurry largely due to hand jitter while taking pictures.
CCPD-Weather	Images taken on a rainy day, snow day or fog day.
CCPD-Challenge	The most challenging images for LPDR to date.
CCPD-NP	Images of new cars without a LP.

Table 3. Descriptions of different sub-datasets in CCPD.

The images are labeled with the following notation:

LP number. Each image in CCPD has a single LP. Each LP number has a Chinese character, a letter, and five letters or numbers. The LP number is an important metric overall.

LP bounding box. The bounding box label contains the coordinates of the initial bounding box. These two points can be utilized to locate the minimum bounding rectangle of LP.

Four vertices locations, contains the exact (x, y) coordinates of the four vertices of LP in the whole image. As the shape of the LP is basically a quadrilateral, these vertices location can accurately represent the borders of the LP for object segmentation.

Horizontal tilt degree and vertical tilt degree, the angle between LP and the horizontal line. After the 2D rotation, the vertical tilt degree is the angle between the left border line of LP and the horizontal line.

Also the area, the degree of brightness and the degree of vagueness.

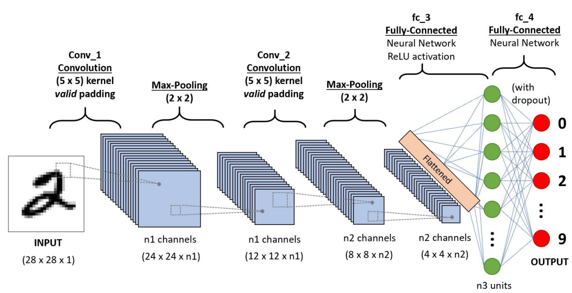
4. Main Algorithm

4.1. Structure

The first component of the structure will be a deep CNN which will identify feature maps from the images. The second component of the structure will be Region of Interest (ROI) pooling, which will extract the most relevant feature maps and classifiers to accurately predict the LP number.

The first module has ten convolution layers. These layers identify both location and character shape. Lower layers also matter because the contain location and shape detail that higher layers may omit. For a feature layer whose size is mxn, the bounding box layer is (m*h)*(n*w). The sizes of m and n are 122, 63, and 33.

When the number of layers increases, the feature map size will keep decreasing. It is easier to predict the bounding box location at higher feature maps. The center point of the bounding box must be located inside the image, and the width and height of the bounding box must be smaller than the width and height of the image. Additionally, multiple features from lower layers are extracted and combined together, as they have more detail than higher layers. Additionally, the area of the LP in these layers is far smaller than the image area. [6]



4.2. Training Methods

One evaluation method is the training objective, which can be divided into the localization loss (loc) and the classification loss (cls). The loc is the smooth loss between the predicted and ground truth boxes. The cls is a cross-entropy loss, which determines the product between the true distribution and the estimated distribution. The algorithm is described by the following equations: [7]

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v), \quad (1)$$

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i), \quad (2)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \quad (3)$$

Additionally, there must be a reasonable initial bounding box prediction. The detection module will be trained from scratch, as there are already numerous types of images. Additionally, the focus will be entirely on the license plate.

5. Coding Methods

I decided to use PyTorch 0.3.1, which is a machine learning library. I initially resized the images to 360x360, and used convolution, batch normalization, and max pooling for ten layers. I set the dropout level to 0.2. I then performed a linear transformation on the variables. I set the data length to 53248. Only western characters are printed, as I was not able to properly render Chinese characters.

When I loaded the images, I resized them and normalized them to a maximum value of 128 to make data analysis easier. I made sure to select every image in the path. When doing forward and backward iterations, I made sure to select the tensor mean each time during the average consensus.

For the training models, I also ran 10 convolutions on each image. I analyzed the data from each layer, as location of the image also counts. During training, the dropout parameter was only set to 0.2 as the entire dataset is much larger. The program was run for ten epochs.

6. Evaluation Metrics

As mentioned in the introduction section, the CCPD dataset is split into several categories based on the defining features of the image. The accuracy for each category will be determined and categorized in a table. Additionally, both detection and recognition methods will be compared to other existing methods such as YOLO and Cascade.

As mentioned before, there are approximately 200k images, which need to be divided into a training and test set. The division splits them into equal amounts, to provide adequate coverage. A .pth model is created as a result.

There will be several sub-components of accuracy measurements. The detection accuracy metric will focus on how close the bounding box is to the license plate. The Intersection-over-Union (IoU) must be greater than 70% to be considered accurate. The recognition accuracy metric will be classified as accurate if the IoU is at least 50%, and if every LP character is correct.

Finally, as there are multiple convolutional layers, each layer will be separately analyzed to determine how overall performance is affected.

From the test runs that were performed, all the models did not perform adequately on CCPD-Rotate, CCPD-Weather, and mostly CCPD-Challenge. Difficulties of detection and recognition on these sets result from the lack of LPs under these conditions in training data. Sub-categories will be important in the future.

7. Conclusion

In conclusion, running the code on the training and testing datasets took far more time than I expected. This is because of the large amount of images that were used. One major issue is that the model was only trained on Chinese license plates, but may not necessarily apply to different kinds of license plates. Additionally, there should have been more cases of license plates in less visible and difficult to detect conditions. Overall, this dataset will most likely prove to be influential in the future.

References

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (alpr): A state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, p. 311–325, Jun 2012.

[2] A. Puranic, D. K., and U. V., "Vehicle number plate recognition system: A literature review and implementation using template matching," *International Journal of Computer Applications*, vol. 134, no. 1, p. 12–16, 2016.

[3] "Plate recognizer." Available at https://platerecognizer.com/ number-plate-datasets/.

[4] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] J. Tian, R. Wang, G. Wang, J. Liu, and Y. Xia, "A two-stage character segmentation method for chinese license plate," *Computers & Electrical Engineering*, vol. 46, p. 539–553, 2015.

[6] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, p. 1476–1481, Jan 2017.

[7] R. Girshick, "Fast r-cnn," in *International Conference on Computer Vision (ICCV)*, 2015.