License Plate Detection

Nicola Farina, Luca Zardini
University of Trento
Department of Information Engineering and Computer Science
AIS - Signal Image and Video

1. Introduction

The problem we had to solve was to develop a program for the detection of license plates in images. By *detection* we just mean drawing a bounding box around the license plate in the original image, without recognizing the characters of the plate.

The solution should work on images taken "in the wild": under different lighting conditions, with no fixed camera (license plates could be distorted by perspective), and with zero, one or multiple license plates in a single image.

Since no particular dataset was provided, we collected two ready-made datasets from the internet:

- **OIDv4** [1]: stands for *Open Image Dataset*, it consists of 6871 images split in training, validation and test and corresponding license plate labels. All the images fit the "in the wild" requirement.

- **AOLP** [3]: stands for *Application-Oriented License Plate Recognition*, it consists of 2049 images, taken in three contexts (more or less controlled and predictable). We took a subset of images as an "easy" dataset, which contains images where the license plates are relatively centered and similar.

2. Proposed Methods

We collected and analyzed several papers regarding the problem at hand, in order to find the current state of the art. We have identified three main approaches:

- **image processing**: numerous papers make use of traditional image processing techniques and algorithms [5][7][2]. Most of them are not recent, as the attention has moved towards machine learning and deep learning;

- **deep learning**: most recent works make use of neural networks, the current state of the art for object detection [4][6][2];

- **mixed methods**: some papers combine image processing and deep learning to gain the advantages of both worlds.

We considered and experimented with all of the previous approaches, and implemented the first two. We discuss them in the following sections.

2.1. Image Processing

Our approach aims at finding contours and filtering them based on properties that are typical of license plates. The main steps are as follows.

2.1.1 Preprocessing

We read an RGB image and convert it to grayscale for reducing complexity and for applying edge detection algorithms. We remove noise using bilateral filtering, which preserves edges. We then apply contrast enhancement using *CLAHE (Contrast Limited Adaptive Histogram Equalization)*.

2.1.2 Edge and contours detection

We find edges in the image using the Canny algorithm. We then apply a closing morphological operation to try and merge adjacent edges. We find contours from these edges, and approximate each contour with a rectangular box. Finally, we filter out boxes that are not rich in white color, have a too small or too large area, or don't have the aspect ratio of a license plate ($width > height$);

2.1.3 Sobel for finding best candidate box

To find the most probable candidate box for a license plate, we use the Sobel algorithm, on vertical edges, to compute the magnitude of each box, and take the one with the highest magnitude. The magnitude is a measure of how many vertical edges there are in the box and how intense they are, under the assumption that license plates have a high magnitude due to the letters and numbers that contrast with

[1] `https://github.com/EscVM/OIDv4_ToolKit`

the backplate. We have chosen the vertical direction to discriminate against the many horizontal edges that cars have (mostly bumpers and vents) and because letters and numbers have more vertical edges.

2.1.4 Results

We have made several assumptions to try to make this approach work decently:

- that license plates are mostly white (not true in some countries);
- that their area falls in a given range (this highly varies if images are taken from different perspectives);
- that there is enough contrast for the edges of the license plate to be detected (depends too much on lighting conditions and the color of the car);
- that there is only one license plate per image (against the initial requirements).

For these reasons, we were not surprised to obtain very poor results: we only managed to obtain an average detection accuracy of 52% on what we consider an easy dataset, with even worse accuracy if the fitting of the bounding box against the license plate is taken into account.

2.2. YOLO

A more promising approach was the usage of neural networks as object detectors. Numerous papers describe the YOLO family as one of the most accurate neural networks used for object detection tasks. All YOLO models are provided with pre-trained weights over the *COCO* dataset, which contains 328k images.

We adopted the latest version, YOLOv5 [2], which was released in June 2020 and updated in October 2021. Among the five available models, we chose the lightest and fastest one, YOLOv5 nano, because of lower training time and because we only need to detect one class.

We downloaded a pre-trained YOLOv5 model and wrote a Python script to train it on the OIDv4 dataset. We ran the training for 100 epochs, with stable performance and little-to-no improvement after a certain number of epochs.

We then wrote a series of Python scripts to wrap the model and provide an interface to validate the model on unseen images using our custom weights.

2.2.1 Results

We tested the trained YOLOv5 net using the testing split of the OIDv4 dataset. The model achieved a score of over 0.9 on precision and recall and a *mAP@[.5:.95]* of around 0.68.

[2]https://github.com/ultralytics/yolov5

The model draws a bounding box over the license plates and assigns them a certain confidence probability.

This type of neural network is not limited by the number of objects in the image, so it can detect more than one plate per image, as per requirements.

2.3. Other Approaches

We have experimented with two other approaches, which we have not published because their performance was not up to par: a Haar Cascade classifier [1]; and mixing image processing techniques with a text detector to find which bounding boxes were license plates.

The **Haar Cascade** approach consists in detecting objects using cascade classifiers based on Haar features. It is a machine learning-based approach where a cascade function is trained from a lot of positive and negative images.

For the **mixed text detector** approach, we chose the *EAST* [3] *(Efficient and Accurate Scene Text)* detector, which is one of the most popular neural networks used for text detection. We downloaded the pre-trained model and we integrated it with our image processing approach. We encountered several problems:

- the traditional image processing struggled to find contours of the plate;
- the neural network often detected text in places where there was no text.

3. Comparison

In this section, we provide some images detected by the two implemented methods:

In Figure 1, the license plate is not detected due to the low contrast between the license plate and the car (both are white), which makes edge detection ineffective. YOLO works fine in such cases 2.

Figure 1: Processing

Figure 2: YOLO

Figure 3 shows another problem encountered with the application of image processing: many times, plates are only partially detected. This problem arises due to low contrast and license numbers and letters not being uniformly spaced.

[3]https://github.com/argman/EAST

Figure 3: Processing

Figure 4: YOLO

Figure 5 shows how, under optimal conditions such as high contrast and front-facing camera, image processing techniques can be valid.

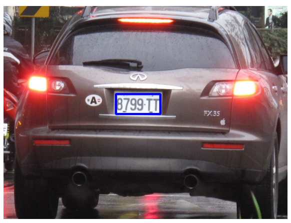

Figure 5: Processing

Figure 6: YOLO

Finally, Figure 7 shows that the YOLO model has high performance also in real traffic situations; it can detect multiple license plates that sometimes are very small.

Figure 7: YOLO

4. Conclusions

We have analyzed numerous papers regarding the problem to solve, found out the best and most common approaches, implemented two different methods for license plate recognition using both image processing methods seen in class and neural network and analyzed them in order to understand which of them is the best and the pros and cons of each of them.

As the images show, it is clear that applying simple image processing on general images has poor performance; it is possible to increase the accuracy by making some strong assumptions, which may be applicable in controlled scenarios (for example, fixed cameras for toll access).

On the other hand, neural networks are powerful tools that can learn the correct features from images making the correct assumptions. The drawback of this method is the difficulty in understanding how it works under the hood and having enough labeled data for proper training.

References

[1] Mohammad Mahdi Dehshibi and Rahele Allahverdi. “Persian vehicle license plate recognition using multiclass Adaboost”. In: *International Journal of Computer and Electrical Engineering* 4.3 (2012), p. 355.

[2] Shan Du et al. “Automatic license plate recognition (ALPR): A state-of-the-art review”. In: *IEEE Transactions on circuits and systems for video technology* 23.2 (2012), pp. 311–325.

[3] Gee-Sern Hsu, Jiun-Chang Chen, and Yu-Zu Chung. “Application-oriented license plate recognition”. In: *IEEE transactions on vehicular technology* (2012).

[4] Rayson Laroca et al. “A robust real-time automatic license plate recognition based on the YOLO detector”. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–10.

[5] Fernando Martın and David Borges. “Automatic car plate recognition using partial segmentation algorithm”. In: *IASTED conference on Signal Processing, Pattern Recognition, and Applications (SPPRA)*. Citeseer. 2003.

[6] Thanh-Nga Nguyen and Duc-Dung Nguyen. “A new convolutional architecture for Vietnamese car plate recognition”. In: *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE. 2018, pp. 7–12.

[7] Satadal Saha et al. “License Plate localization from vehicle images: An edge based multi-stage approach”. In: *International Journal of Recent Trends in Engineering* 1.1 (2009), p. 284.