Towards Real-Time Automatic Number Plate Detection: Dots in the Search Space

Chi Zhang

Department of Computer Science and Technology, Zhejiang University

wellyzhangc@zju.edu.cn

Abstract

Automatic Number Plate Detection[1], or ANPR, has long been a traditional problem and is already widely applicable in police forces around the world for traffic control, toll collection and highway law enforcement. And it is never denied that ANPR would also be an essential part for the currently heatedly discussed autopilot system. In this tech-report, solutions towards real-time automatic number plate detection has been searched, implemented and compared. Though traditional methods such as contour-based and SVM-based number plate detection do satisfy the real-time constraint, their accuracy are considerably lower than deep-learning-based approaches[2, 3], which after years of research, have progressed to a stage where powerful computation devices enable faster and faster proposal- and region-based methods, making the real-time constraint achievable. Therefore, solutions to this problem is explored in this report and a periodical conclusion is drawn.

1 Introduction

Vehicle Registration Plates, or number plates, have been used as the unique identification for vehicles and links to the drivers who own them and thus a very efficient way for police forces to collect evidence and apply law enforcement to those related to traffic rule breakers. The detection and recognition problems have already been explored for a long time and some currently used methods involve closed-circuit television, road-rule enforcement cameras and cameras specifically designed for this task[1]. However, the accuracy and efficiency could not be normally realized at the same time and human labor is always required. A widely used pipeline has one speed camera to screen out speed limit and other law breakers, another camera to capture the instance the driver violates the rule and human at the traffic regulation centers to manually recognize the number plates. This process is neither efficient nor convenient. Therefore, an accurate and real-time number plate recognition system is highly desired which could be used both for the city monitoring and recently risen autopilot systems.

Normally, such a system would be divided into two components: number plate detection and digit recognition. My work here focuses completely the number plate detection and extraction from raw images. And it could be summarized as follows:

- Related work is first classified, reviewed and discussed.
- Traditional and advanced approaches on this problem are implemented and explored.
- Algorithms are evaluated on one private dataset for the purpose of comparison and transfered to a new dataset for quality inspection.
- Performance and future work are discussed.
- A periodical conclusion is made to close the report.

2 Related Work

For the detection part of the system, it is completely embedded in the broader field of and a subtask in object detection. This classic and high-level computer vision problem exists for decades and nobody has ever claimed to solve it.

Some traditional methods could be traced back to the *contour-based* learning[4, 5] where the authors, based on the assumption that contours constitute a major part of human recognition for objects, propose to use contours as clues to detect the existence of various objects.

This assumption then gradually evolves to a new phase where contours are replaced by a more general concept of *features*. Computer vision scientists believing that features are characteristic of objects gradually proposed a variety of image features, including but not limited to *HOG*[6], *SIFT*[7] and *SURF*[8]. The combination of features and classifiers then somehow dominated the task of object detection, one typical example of which is HOG+SVM[9]. This algorithm has proven to be both accurate and efficient enough for object detection and this powerful idea also gave birth to numerous variants. Then came the *Deformable Part Model*[10], or DPM. DPM leverages the old idea of part-based model and combines this idea with modern features and machine learning technique to boost the performance of object detection.

Currently, the research community has been focused on the deep learning approaches to this computer vision task after Hinton proved the prospect of deep learning in the ImageNet image recognition task[11]. Ever since then, computer scientists have proposed two categories of algorithms that attempt to solve the detection task. One of them is proposal-based. This category of algorithms first generate region proposals, rather then using sliding windows, and then use the deep neural network as an advanced classifier to predict what the region proposals are[12, 13, 14]. The second category of algorithms equally divide the spatial space of the image and simultaneously predict the class and bounding box of an object in a specific cell, using features extracted by the deep neural network[15, 16]. Both of the categories transform the detection problem into one that combines bounding box prediction and image recognition.

3 Methodology

All of the methods embody the powerful idea of image features combined with machine learning technique. What they differ lie in the image feature extractors and the classifiers.

3.1 Traditional Method

A traditional detection method in the machine learning and computer vision literature traces to one that uses the HOG feature, image pyramids, sliding windows and SVM. Thus this method is tried to solve the number plate detection problem first.

3.1.1 HOG Feature

Histogram of Oriented Gradients, or HOG, is widely used in compute vision tasks. This feature descriptor computes the number of gradient orientation in localized portions of an image of a dense grid with uniformly spaced cells and uses overlapping local contrast normalization to improve the accuracy. Other features such as edge orientation histograms, Scale-Invarient Feature Transformation or shape features are also applied in computer vision tasks but turn out not as effective and accurate for the detection problem.

3.1.2 Sliding Window and Image Pyramid

Given a raw image, it's rather impossible for a machine to automatically focus on a specific portion of the image, due to the lack of the attention mechanism human possess. And one typical way to address this is to use a window much smaller than the spatial dimension of the image and slide it across the image and extract the image features local to the window. This method is thus called sliding window. However, sliding windows do not consider the relative size of an object to the image. A number plate could either take one-third space of the image or half of the space of the image, and therefore a fixed image space and a fixed size of the window would significantly reduce the accuracy of the number plate detector.

Luckily, image pyramids are an effective way to solve this problem. For a single frame captured by the camera, the size of a sliding window is still fixed, but the image is scaled to different sizes, proportional to each other. Now a window could be slided over different scales of the same image, making detection of number plates with different sizes possible.

3.1.3 Support Vector Machine and Hard Negative Mining in Detection

After a window is produced and the features in it are described, a Support Vector Machine is connected. This classifier predicts whether this window contains a number plate: if the prediction is positive, the image patch is picked out and stored for further number plate recognition. But it's highly possible that after the image is scaled to different sizes, there could be multiple overlapping patches with positive prediction. One way used for the task to address this problem is called hard negative mining[17]. The idea of this technique is quite straightforward: negative samples predicted positive should be paid more attention and trained again as negative examples. Therefore, the false positive patches are relabeled again as negative and fed to the model to be trained again. An ingredient that introduces more robustness into the problem treat the distance to the SVM hyperplane as a threshold and only patches far enough from the hyperplane should be considered positive.

3.2 Advanced Methods

Deep learning approaches have proven effective in computer vision tasks, especially in image recognition, object localization and detection[18, 19, 20]. And over the years, the research community has realized that both region-proposal-based methods and cell-based methods are effective in solving the detection problem.

3.2.1 Faster RCNN

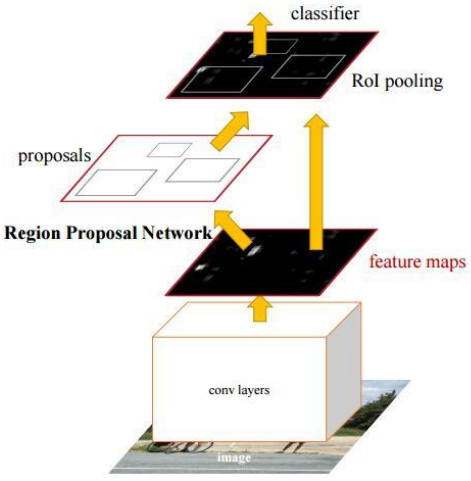

Figure 1. Faster RCNN pipeline.

Faster RCNN[14] is the most advanced algorithms of the kind that is based on region proposal. In this model, a image is fed entirely into a deep neural network. After several layers of feature extraction, features of the image are taken by another network, called

Region Proposal Network[14], which then outputs the bounding boxes of number plates. These proposals are pooled into a fixed size by a ROI pooling layer and then fed to a SoftMax[21] classifier to predict the existence of the number plate.

One advantage of the Faster RCNN model lies in the fact that unlike its predecessors of RCNN[2] and Fast RCNN[13], this model could be trained completely end-to-end and demonstrates its superiority in the region proposal generation speed.

Though Faster RCNN has enjoyed the remarkable boost in detection speed, it still could not be used for applications with the real-time constraint.

3.2.2 YOLO

YOLO[15] then uses an extremely distinctive idea in object detection. Unlike the region-based methods, YOLO uniformly divides the image into cells. In each cell, there could be several object detectors, each of which has the class and the bounding box of one object in the cell. During training, the deep neural network is built, takes images from the training dataset as input, forms a loss function that incorporates the cross-entropy loss, the L2 regression loss and the randomness and backpropogates[22] the gradients to update the parameters. Thus, during testing, the network output contains whether number plates exist and if yes their bounding boxes.

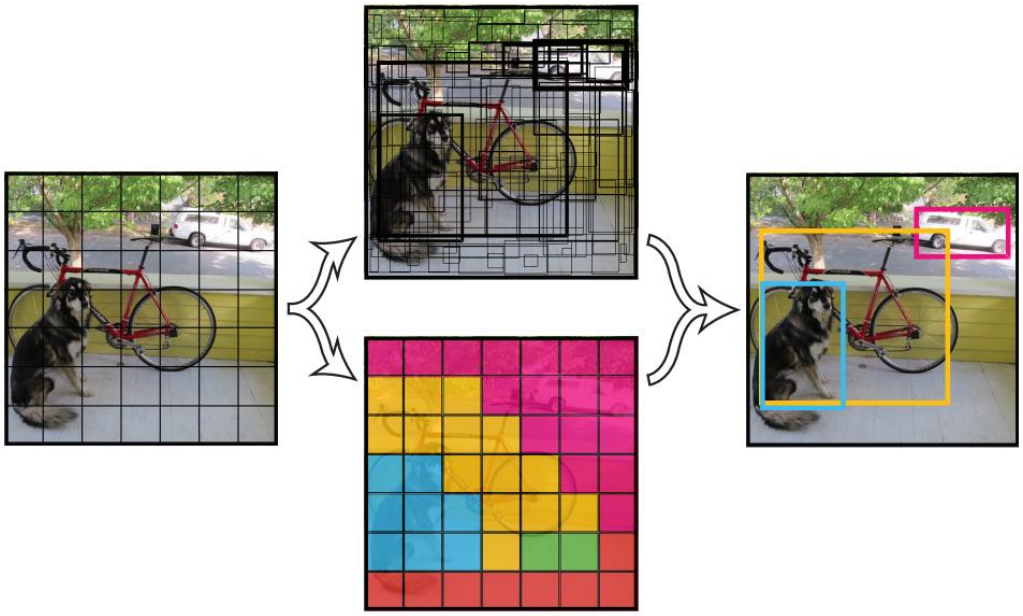

Figure 2. Sketch of the YOLO algorithm.

4 Evaluation

All of the algorithms are evaluated on a private benchmarking dataset. This dataset is

split into a training set and a testing set in a ratio of 9:1; each of the pictures has exactly one number plate in it; pictures are taken either in the day time or at night; orientations and backgrounds of the images vary drastically. Qualitative evaluation of the approaches is also examined by testing the algorithms on a completely different dataset, with test cases having one or more number plates each.

Figure 3. YOLO on a new dataset for qualitative evaluation.

Performance of the three algorithms are listed in the following table.

Model	mAP@0.5	No.Conv	Test Time	Init	Multi-obj
YOLO	**97.7**	9	0.022s	No	Easy
Faster R-CNN	45.0	5	0.077s	Yes	Easy
HOG+SVM	95.8	-	**0.010s**	No	Hard

Table 1. Performance evaluation of the three approaches.

As could be seen above, HOG+SVM achieves a comparable accuracy with YOLO and both of them could be used for real-time applications. However, though HOG+SVM is fast and accurate, it does not handle cases with multiple number plates and therefore might not be useful in the real world. But YOLO on the other hand, is faster, accurate and adaptable to harder cases.

Figure 4. YOLO on the benchmarking dataset.

5 Discussion

As could be seen above, the traditional method of HOG+SVM performs only well enough in the number plate detection. But since it's not end-to-end and requires the relabeling process, its training phase is much more complicated than the two deep learning approaches. Besides, traditional methods does suffer from its inferiority in detection accuracy. This should be rooted in the feature effectiveness, since deep learning automatically extract image features and is better tuned to any specific problem. One serious problem that renders Faster RCNN not as accurate as expected might result from the size of the training set and the inappropriate hyperparameters setting, as the Faster RCNN algorithm has such a large model capacity. YOLO has the best performance over all the other, but a hidden problem not exposed here has to do with the relative size of an object to the image. As the YOLO paper states, it still is not well solved when an object is too small compared to the dimension of the cell it is in.

6 Conclusion

In this tech-report, three major methods in object detection have been implemented and evaluated on the task of number plate detection: HOG+SVM, Faster RCNN and YOLO. As it turns out, HOG+SVM is fast enough for the real-time constraint but not as accurate as YOLO; Faster RCNN is neither as fast nor as accurate as YOLO; YOLO has performed best in both achieving a high accuracy and also satisfying the time constraint. Besides, the YOLO has the best performance when transfered to a completely new dataset after training, validating and testing on the benchmarking dataset.

References

[1] Wikipedia. Automatic Number Plate Recognition, https://en.wikipedia.org/wiki/Automatic_number_plate_recognition

[2] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint*: 1311.2524.

[3] Ross Girshick. Fast R-CNN. *arXiv preprint*: 1504.08083.

[4] Shotton, Jamie, Andrew Blake, and Roberto Cipolla. Contour-based learning for object detection. *Tenth IEEE International Conference on Computer Vision (ICCV'05)* Volume 1. Vol. 1. IEEE, 2005.

[5] Xu, Yong, et al. Contour-based recognition. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

[6] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 886-893). IEEE.

[7] Lowe, David G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60.2 (2004): 91-110.

[8] Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer Berlin Heidelberg.

[9] Hilton Bristow and Simon Lucey. Why do linear SVMs trained on HOG features perform so well? *arXiv preprint*: 1406.2419.

[10] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627-1645.

[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2014, September). Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision* (pp. 346-361). Springer International Publishing.

[13] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440-1448).

[14] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

[15] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once:

Unified, real-time object detection. *arXiv preprint* :1506.02640.

[16] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., & Reed, S. (2015). SSD: Single Shot MultiBox Detector. *arXiv preprint*: 1512.02325.

[17] Henriques, J. F., Carreira, J., Caseiro, R., & Batista, J. (2013). Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *proceedings of the IEEE International Conference on Computer Vision* (pp. 2760-2767).

[18] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint*: 1312.6229.

[19] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9).

[20] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint*: 1512.03385.

[21] Wikipedia. Softmax Function. URL: https://en.wikipedia.org/wiki/Softmax_function.

[22] LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9-48). Springer Berlin Heidelberg.