Robust License Plate Detection In The Wild

Gee-Sern Hsu
Artificial Vision Laboratory, Dept. of Mechanical Engineering,
National Taiwan University of Science and Technology, Taipei City, Taiwan 106.
jison@mail.ntust.edu.tw

ArulMurugan Ambikapathi
Utechzone Co. Ltd.,
New Taipei City, Taiwan 235.
aareul@ieee.org

Sheng-Luen Chung, Cheng-Po Su
Dept. of Electrical Engineering,
National Taiwan University of Science and Technology, Taipei City, Taiwan 106.
slchung@mail.ntust.edu.tw

Abstract

License Plate Detection (LPD) is the pivotal step for License Plate Recognition. In this work, we explore and customize state-of-the-art detection approaches for exclusively handling the LPD in the wild. In-the-wild LPD considers license plates captured in challenging conditions caused by bad weathers, lighting, traffics, and other factors. As conventional methods failed to handle these inevitable conditions, we explore the latest deep learning based detectors, namely YOLO (You-Only-Look-Once) and its variant YOLO-9000 (referred here as YOLO-2), and customize them for effectively handling the LPD. The prime customizations include modification of the grid size and of the bounding box parameter estimation, and the composition of a more challenging AOLPE (Application-Oriented License Plate Extended) database for performance evaluation. The AOLPE database is an extended version of the AOLP database [1] with additional images taken under extreme but frequently-encountered conditions. As the original YOLO and YOLO-2 are not designed for the LPD, they failed to handle the LPD on the AOLPE without the customizations. This study can be one of the pioneering works that revise state-of-the-art real-time deep networks for handling the LPD. It also serves as a case study for those who wish to customize existing deep networks for detecting specific objects. In addition to a pioneering explorations of deep networks for handling the in-the-wild LPD, our contribution also includes the release of the AOLPE database and evaluation protocol for a novel benchmark for the LPD.

1. Introduction

License plate detection and recognition (LPDR) is one of the vital fields of computer vision and has been widely investigated in the recent years, owing to its high practical importance [1, 2]. LPDR applications include areas that can be categorized as access control, law enforcement, road patrol monitoring, and many more [1]. It is well-known that the LPDR includes two phases, namely: (i) License Plate Detection (LPD) phase and (ii) License Plate Recognition (LPR) phase, and the accuracy of the detection phase has a direct impact in dictating the efficacy of the recognition phase. This work is focused on the first phase i.e., design and development of LPD algorithms. While a plethora of reliable algorithms is available for LPD under controlled environments, in-the-wild LPD algorithms that are robust to inevitable practical issues such as occlusions, changes in illumination, shadow effects, varying camera view point, variation in distance between camera and license plate, etc., are still currently being developed and are yet to reach the desired accuracy required for practical and reliable real-time implementations [1, and references therein]. Most of the existing and recently developed LPD algorithms are still based on image enhancement techniques and hand-crafted features [3, 4]. The current limitations of such LPD algorithms may be attributed to the lack of robust hand-crafted features that can account for severe variations in the imag-

978-1-5386-2939-0/17/$31.00 ©2017 IEEE

ing parameters, which severely inhibits the detection accuracy, especially for license plates in the wild (images captured under uncontrolled but practically feasible lighting and imaging environments). The computation complexity of existing LPD algorithms is also a prime concern as conventional LPD algorithms take a significantly longer portion of the total computational load of an LPDR system. As a consequence, most of the existing LPD algorithms are not suitable for practical applications, due to both limited efficacy and higher computational requirements.

Recently, deep learning approaches have dominated and yielded excellent state-of-the-art results, which were considered a distant reality, in various fields of image classifications. Nevertheless, the application of deep learning for developing robust LPD algorithms, is still in its very initial stage in the LPDR research community [5]. This work aims in bridging this gap and can be regarded as one of the first solid contribution in this direction. Recent advancements in deep learning based multi-object detection includes YOLO (You-Only-Look-Once) [6] and most recently, the variant of YOLO, namely YOLO-9000 (referred as YOLO-2, for convenience) [7]. Unlike conventional detection methods that are based on classifiers for detection, the YOLO method [6] models the object detection as a deep learning based approach and aim to predict the bounding boxes and the associated class probabilities. Other famous deep learning based detectors include R-CNN [8], Fast R-CNN [9], and faster R-CNN [10]. Both YOLO and YOLO-2 are the most faster and reliable detectors than their contemporaries and hence are much suitable for real time object detection. While these algorithms are effective and efficient in detecting a wide class of objects, they failed miserably when applied directly for LPD (with basic LPD training). The apparent reasons for such failures include, challenging lighting conditions, shadowing effects, and several other factors. Inspired by the prowess of deep learning in computer vision, especially in the field of object detection and tracking, in this work we have endeavored to customize these state-of-the-art deep learning based detection algorithms exclusively for LPD. To the best of our knowledge, this work is one of the first to explicitly address the robust and practical means of LPD using the novel deep learning detection approaches. We believe that this work will also serve as a platform for various researchers by demonstrating the customization of deep learning approaches according to their tasks and requirements (LPD, in our case).

On the other hand, in any computer vision applications, fair evaluation on a solid and standard benchmark database is a mandatory requirement, both for researchers and users. Especially, with the recent advent of deep learning in almost all fields of computer vision, the demand for an exhaustive and comprehensive database can never be undermined. However, for LPD there are very limited database available and this includes the database from Anagnostopoulous et al. [11], and our recently introduced Application-Oriented License Plate (AOLP) database. Though the latter contains more images under different categories, the need and demand for a larger database with more realistic and challenging environments, motivated us to develop an extended database, namely AOLP extended (AOLPE), with more categories and more image sequences under each category. It should be mentioned that the AOLPE is by far the most comprehensive database in the LPD and LPR research community that will be made publicly available, along with the benchmark results and evaluation protocols. Detailed descriptions of the AOLPE database will be presented in later sections. The customized deep learning based detectors that are proposed in this work, are evaluated on the extensive AOLPE database to demonstrate their superior performances over other contemporary approaches. Thus, the prime contributions of this work include: (i) Customization of state-of-the-art detection algorithms, namely YOLO, and YOLO-2, exclusively for LPD, (ii) Creation of an extensive AOLPE database, and (iii) Rigorous validation of the customized algorithms (and their predecessors) on the AOLPE database.

The rest of the paper is organized as follows: The Application-Oriented License Plate Extended (AOLPE) Database is presented in Section 2, where the merits of the currently introduced database is discussed in comparison with its predecessor. The state-of-the-art deep learning based detection algorithms and their simple yet powerful customizations for LPD are discussed in Section 3. A detailed experimental analysis, results, and the associated discussions are presented in Section 4. Finally, concluding remarks and future directions are drawn in Section 5.

2. The Application-Oriented License Plate Extended (AOLPE) Database

In the current scenario of computer vision algorithm developments, it will not be an over exaggeration to state that the presence of a benchmark database is as important as the design of standard relevant algorithms. The lack of a standard database for LPD is a long-standing issue, briefly circumvented by the introduction of AOLP database in 2013 [1]. The variables contained in the AOLP database are rotation (in yaw, pitch and roll), size and illumination conditions. However, as autonomous driving is considered one of the central concerns in the vehicular technology, a desired LPD solution must be able to handle 1) different weather conditions; 2) difficult lighting conditions, for example, headlight glare or against the lights in general; 3) multiple plates; 4) partial occlusions, and other variables that one would encounter during driving. Hence, the requirement for a more exhaustive benchmark database that includes samples under the above mentioned extreme but practical conditions, has become a need of the hour for fair validation of LPD algorithms. We have, therefore, collected a new set of data from various conditions that cover all these variables and merged this dataset with the original one, and named the whole collection as the Application Oriented License Plate Extended (AOLPE) database. In addition to the AOLP, samples under the following realistic conditions are also included in the AOLPE database:

- Extreme Weather Conditions: The AOLPE database contains samples under different weather conditions. For instance, images with direct sunshine, images during rainy conditions, and other variations.

- Scene Complexity: In reality, one may expect the license plate of a vehicle be amidst different backgrounds. This includes other vehicles (relevant or irrelevant) at the background, shades or shadows from other vehicles or nearby objects such as road signs, etc.

- Glaring / Lighting Effects: In the night or in the tunnel, often the light glares from the other oncoming vehicles affect the clarity of the license plate captured. But this is a very common aspect and hence needed to be taken into account.

As one could witness, the above realistic conditions are the ones normally encountered and any LPD algorithm should be made robust to these conditions and therefore can substantiate its practical applicability. Sample illustrations for the above scenarios are depicted in Figs. 1 - 3. It could be readily observed that most of the scenarios depicted in Figs. 1 - 3 are naturally common and from an algorithmic perspective, they are much difficult to handle, as the license plates are hardly visible in these circumstances. In total, the AOLPE database contains more than 4200 images, under different imaging and lighting conditions. It should be emphasized that these images are license plate images taken at random conditions / backgrounds and do not hold any intended background correlation among themselves.

Figure 1. Some samples from the AOLPE database showing the variation in day light conditions.

In Section 4, the proposed customized versions of the LPD will be evaluated on the AOLPE database, to demonstrate their efficacy under the above mentioned practically *wild* situations.

Figure 2. Some samples from the AOLPE database showing the effect caused by head light glares and dark lighting conditions.

Figure 3. Some samples from the AOLPE database illustrating the imaging artifacts caused due to rainy conditions.

3. Customization of State-of-the-Art Object Detectors for License Plate Detection

In this section we will begin by briefly discussing about the state-of-the-art deep learning based detectors, namely YOLO [6], and YOLO-2 [7], and proceed to expose their limitations for LPD. We then explain the customization techniques applied to these algorithms, so as to precisely identify the license plates under various lighting conditions and imaging environments.

3.1. Brief Review of YOLO and YOLO-2, and their Limitations

Deep learning based detection algorithms have recently outperformed the conventional detection algorithms, by a great margin. The most recent state-of-the-art algorithms namely YOLO [6], and YOLO-2 [7], are currently the benchmarked detection algorithms and are shown to yield better performance coupled with reduced computational cost, than most other contemporary methods, and therefore constitute the central theme of this work. Unlike its predecessors that used specially designed classifiers for detection, the YOLO trains a single deep learning network and aims to predict the bounding boxes and the associated class probabilities directly from the full image frame. Precisely, YOLO considers the entire image, uniformly divided into multiple patches, and uses regression analysis to detect an object. The class-specific confidence scores in each patch are calculated at the fully connected layer of the modified GoogleNet architecture, and are finally used for object detection. For complete details on the architecture, activation function, training procedures, and precise definitions of multi-box class scores, please refer to [6]. It has been shown that the base YOLO architecture can process at 45 frames per second (FPS) and achieve mean average precision (mAP) of 63.4. A faster version, namely Fast YOLO can process around 155 FPS, but with a compromised mAP of 52.7 (still better than many other detection methods).

However, though YOLO is much faster than methods like faster R-CNN [10], it results in more localization errors when detecting smaller objects [6] and thus making it unsuitable for LPD.

It is also worth to refer to the very recent extension of YOLO, namely YOLO-2 [7]. In [7], the authors demonstrated that the performance of YOLO-2 is by far the best when compared to other existing approaches. The YOLO-2 can detect around 9000 objects with a higher mean average precision than YOLO and others, and can process at a higher frame rate. The YOLO-2 utilizes a modified deep learning CNN namely, Darknet-19, along with improvement techniques for variable anchor boxes and multi-scale training. There are a series of techniques proposed and discussed in [7], for improving the performance of the detector.

Hence, the current requirement for the utilization of YOLO, and YOLO-2 for LPD is to customize these algorithms to yield better performance for LPD, without compromising on their computational speed. Before proceeding with the customized extensions of these detectors, it is worth to briefly discuss the generalized loss functions of these networks. The loss functions form the core of the training methodology, which dictates the learning and testing accuracy of the detectors. The training phase is therefore a methodology to make the network learn by effectively and efficiently minimizing these loss functions (aka cost functions). Though the above detectors differ in their architecture, training methodologies, performance etc., their loss function can be generalized as follows:

$$L(pos, wd, ht, conf) = L_{localization}(\cdot) + L_{confidence}(\cdot), \tag{1}$$

where wd and ht corresponds to the width and height of the bounding boxes of the detected target, pos corresponds to the central position of the target, and $conf$ represents the confidence score. The confidence score should be high when the object is present in that localization, else it should be zero. As can be observed from the Equation (1), the loss function constitute of two functions: $L_{localization}(\cdot)$ controls the localization error and the other function $L_{confidence}(\cdot)$ focuses on the confidence score. During training, these parameters $(pos, wd, ht, conf)$ are to be provided for each and every training image and the corresponding functional values are calculated at the fully connected layer of the respective deep learning networks. The training phase involves the update of the weights associated in the CNN, such that the overall loss is minimized. In general, the well-known stochastic gradient descent (SGA) algorithm, with appropriate learning parameters are used to update the weights until convergence.

3.2. Customized Extension of YOLO and YOLO-2

After exploring the potential limitations on applying the afore mentioned benchmark detectors for LPD, in this section, we will propose and discuss the relevant customization techniques to the respective detection algorithms.

For the customized YOLO (named as YOLO-LPD, for convenience), we begin by modifying the fully connected layers of the YOLO GoogleNET, which was designed for dividing the input image using 7x7 grid (49 patches). Consequently, the YOLO fails to detect smaller objects. Quite different from YOLO that is designed for multiple-class objects detection, for YOLO-LPD the number of classes is only two: one is the license plate and the other is the background. Furthermore, given the fact that the size of license plates in each image might be different, and there could be serious issues when more than one license plate occur in the same grid (this will be the case when more vehicles, especially two-wheelers, are captured in close proximity), we have opted for a larger number of patches with a 11x11 grid structure, and hence redesigned the fully connected layer accordingly. Such a 11x11 grid will ensure that the small license plates are also detected. The network is also remodeled such that each grid only can predict one bounding box (to avoid false positives), and to improve the bounding box error when the license plate in the image is smaller than the size of the grid. It is worth to mention that we have also resorted to a higher dimensional grid to accommodate detection of much smaller plates. However, this resulted in more classification errors and false alarms. Hence, we have finalized on employing the 11x11 grid for YOLO-LPD. For training the modified GoogleNet architecture, we employed the original YOLO pre-trained model (as initial weights), which already has been proved to have the ability to detect larger objects. Since, the original YOLO has been trained with the different database, in this work we have used our proposed AOLPE license plate images to train the customized GoogleNET network specifically for LPD. An overall illustration of the YOLO-LPD network is shown in Fig. 4. A similar modification has been devised for YOLO-2, as well. The testing of trained YOLO-LPD and YOLO-2-LPD networks, and their performance comparisons will be detailed in Section 4.

4. Performance Evaluation and Discussion

In this section, we perform various experiments to demonstrate the efficacy of the proposed customized extensions of the above mentioned detection algorithms, exclusively for LPD. The experiments are conducted with the more rigorous and complete AOLPE database, presented in Section 2. It should be mentioned that the respective original detection algorithms (without any modifications) almost had negligible detection ability for identifying the

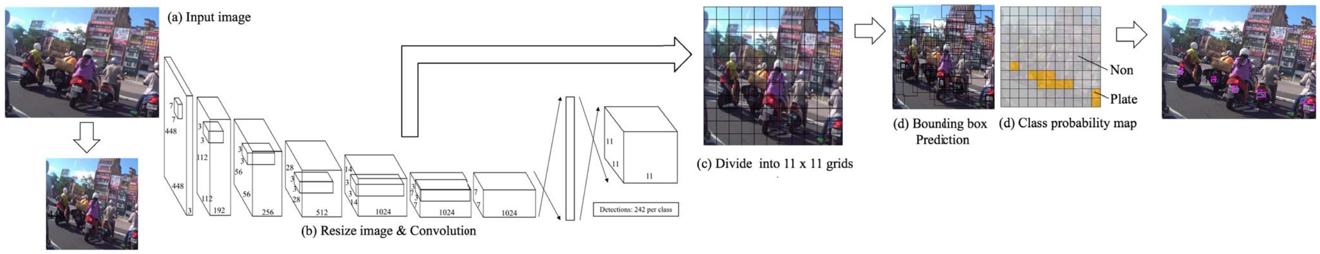

Figure 4. Illustration of the stages in YOLO-LPD.

license plates. Therefore, the methods under consideration includes the customized counterparts, namely YOLO-LPD, and YOLO-2-LPD. Since the above algorithms are proved to exhibit the best detection performance when compared to their contemporary approaches such as R-CNN [8], Fast R-CNN [9], and faster R-CNN [10], the latter are not considered in this performance comparison. For detection accuracy the mean average precision (mAP) is generally used for quantifying the accuracy. However, since the number of classes in LPD is just one (i.e., license plates, while the other class is nothing but the background), we use average precision as the performance measure. The average precision is directly related to the area under the precision-recall curve and hence, we will mainly rely on the precision-recall curves for a visual performance comparison. It should be noted that the larger the area under the precision-recall curve, the higher the average precision, and better is the detection performance. For the sake of fair comparison, all the experiments have been performed using Ubuntu OS based PC, with Intel Core i7 - 3.4 GHz CPU, with a RAM of 16GB, and with GeForce GTX TITAN X GPU. For training and testing the customized detectors, a well-known open source, namely the Caffe framework [12] has been employed.

The parameter settings for the customized detectors are summarized as follows. The learning rate, momentum, decay rate, and maximum number of iterations, are empirically set as 0.001, 0.9, 0.0005, and 40000, respectively, for all the customized LPDs. For YOLO-LPD and YOLO-2-LPD, the batch size is set to 64 and leaky rectified linear unit (ReLU) is used as the activation function. For the sake of considering more samples for training, we have also used 1200, randomly browsed and chosen license plate images from the world wide web. These additional images are referred to as AUX (auxiliary) database, for convenience. In the sequel, experiments will confirm that the extra 1200 images are redundant, and the current images in the AOLPE database are sufficiently large for training deep learning based LPDs. The total number of images under each category of the AOLPE database, and the split up for training and testing are given in Table 1.

Basically, there are four experimental setups considered in this performance evaluation of the proposed customized LPDs. For the total number of images used for training and testing under each experimental scenario, please refer to Table 1. The first experiment is performed by considering only the AOLP training set (300 images) and AUX database (1200 images), for training. The second experiment involved the recently added images of the AOLPE training set of extension part (1661 images) and AUX, for training. While the third experiment considers all the AOLP, AOLPE, and AUX databases for training, the fourth experiment just considers only the AOLPE database, for training. For the sake of a fair statistical inference, in all the four cases, both the original AOLP images and the AOLPE images (in the wild) are used for testing. The precision-recall curves for the customized LPDs, for the four experimental setups are given in Fig. 5. It can be observed from Fig. 5 that the performance of the customized YOLO-2-LPD is not satisfactory indicating that the training performed by AOLP and AUX are not sufficient to detect the license plates in the wild. Also, it is obvious to note that for all the experiments, YOLO-2-LPD exhibits better performance than YOLO-LPD. The dashed line (Exp. II), arrowed line (Exp. III), and dashed-dotted line (Exp. IV) show that the detection performance of the customized YOLO-2-LPDs are fair enough, as AOLPE is involved in all these three cases. It is important to note from Fig. 5 that the accuracy of the LPDs are at their best even when AOLPE is considered alone. This proves the sufficiency of in-the-wild images in the AOLPE database, that can account for a wide range of imaging scenarios in realistic situations. The above experiments indicates that partial training with our new AOLPE database is sufficient for LPD in all aspects, as it can single-handedly yield the best detection performance for the customized LPDs.

As far as the computational complexities of the LPD algorithms are concerned, it is expected that the customized versions should be faster than their respective predecessors, as some of the steps are simplified in the customized versions according to the nature of license plates. While the original YOLO had a detection frame rate of 45 fps, the customized version reach a detection rate of 54 fps. Obviously, the frame rate improvement is not so significantly higher and this could be partially attributed to the increase in the grid size (from 7x7 to 11x11) for the YOLO-LPD.

Table 1. Split up details of the number of images under each category of the AOLPE database, for Training and Testing.

AOLPE Database						
	AOLP			Extension		
Category	Access Control [1]	Law Enforcement [1]	Road Patrol [1]	Day Conditions	Night Conditions	Wet Conditions
Train	100	100	100	673	503	485
Test	581	657	511	150	150	200
Total	681	757	611	823	653	685

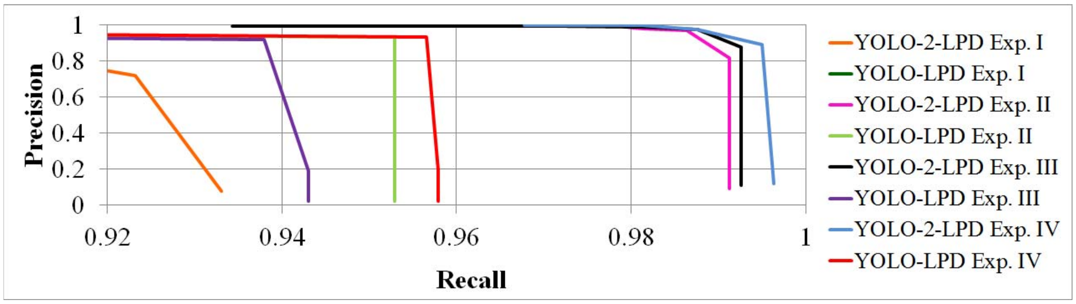

Figure 5. Precision-Recall curves for the customized LPDs for Experiments I-IV. Note that the performance of YOLO-LPD in Exp. I (dark green curve) is far worse than others and hence not shown in this graph for the sake of clarity.

5. Conclusion

LPD is one of the most challenging phase of LPDR as its accuracy and complexity determines the overall performance of the LPDR system. In this work, we have studied the most recent state-of-the-art detection algorithms and exposed their limitations when applied for LPD. Accordingly, we have customized the respective detection algorithms exclusively for LPD and demonstrated that the modified versions perform better. To the best of our knowledge, this is the first work that addresses the utilization of deep learning based detectors exclusively for LPD. Most importantly, a complex and exhaustive database, namely AOLPE has been introduced to the LPD research community, and benchmark results are obtained by using the customized LPD detection algorithms. We believe that these benchmark standards and the AOLPE database will have a huge impact in the LPD research community, by serving as a principal protocol for evaluating the efficacy and practical applicability of LPD algorithms. Also, the customization approach presented in this work would incubate further researches in other related domains. As can be witnessed from Fig. 5, there is still room for improving the precision-recall curves of the customized LPDs. The increase in detection accuracy can be possibly achieved by, including more training samples (i.e. increasing the size of the AOLPE database), opting for more deeper and complex deep learning networks, and more rigorous analysis of the multi-scale feature maps, anchor boxes, and aspect ratios. These performance boosting techniques are currently under investigation. Another interesting future directions would be to smartly combine the conventional hand-crafted features based LPD with the deep learning training mechanisms, to achieve better accuracy.

References

[1] G. S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE TVT*, vol. 62, no. 2, pp. 552–561, Feb. 2013.

[2] Y. Wen *et al.*, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 830–845, Mar. 2011.

[3] V. Tadic *et al.*, "Fuzzified gabor filter for license plate detection," *Eng. App. of A. I*, vol. 48, pp. 40–58, 2016.

[4] S. Azam *et al.*, "Automatic license plate detection in hazardous condition," *Vis. Com.. I. R.*, pp. 172–186, 2016.

[5] Y. Liu *et al.*, "Convolutional neural networks-based intelligent recognition of chinese license plates," *S. Comp.*, pp. 1–17, Feb. 2017.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, realtime object detection," *arXiv:1506.02640*, 2015.

[7] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv:1612.08242*, 2016.

[8] R. Girshick *et al.*, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, June 2014.

[9] R. Girshick, "Fast r-cnn," in *Proc. ICCV*, Dec. 2015.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, Dec. 2015.

[11] C. N. Anagnostopoulos *et al.*, "License plate recognition from still images and video sequences: A survey," *IEEE T.. Intel. Transp. Syst.*, vol. 9, no. 3, pp. 377–391, Sep. 2008.

[12] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.