AWFA-LPD: Adaptive Weight Feature Aggregation for Multi-frame License Plate Detection

Xiaochang Lu
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Yuan Yuan
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Qi Wang[*]
crabwq@gmail.com
Northwestern Polytechnical University
Xi'an, Shaanxi, China

ABSTRACT

For license plate detection (LPD), most of the existing work is based on images as input. If these algorithms can be applied to multiple frames or videos, they can be adapted to more complex unconstrained scenes. In this paper, we propose a LPD framework for detecting license plates in multiple frames or videos, called AWFA-LPD, which effectively integrates the features of nearby frames. Compared with image based detection models, our network integrates optical flow extraction module, which can propagate the features of local frames and fuse with the reference frame. Moreover, we concatenate a non-link suppression module after the detection results to post-process the bounding boxes. Extensive experiments demonstrate the effectiveness and efficiency of our framework.

KEYWORDS

License plate detection, optical flow, multi-frame, feature aggregation, adaptive weight

1 INTRODUCTION

License plate detection (LPD), locating the license plates (LPs) in images, is a crucial branch of the intelligent transportation systems (ITS). LPD is an integral stage of a complete Automatic License Plate Recognition (ALPR) system, which could be widely used in a variety of applications such as self-driving systems, parking fee management, and road traffic monitoring [7, 13, 15].

With the rapid development of deep neural networks in the past few years, numerous work has been applied to ALPR systems. A complete ALPR system generally contains two stages: firstly locate the LPs by object detection methods and crop the area; secondly identify the LP characters through text recognition algorithms. Identifying the LP numbers in the cropped regions could be simply divided into two categories, segmentation based methods and segmentation-free methods. In [7], Li *et al.* recognize the LP characters by adopting recurrent neural networks (RNNs) and connectionist temporal classification (CTC), without character segmentation. In [5], Laroca *et al.* utilize the CR-NET [12], which combines character segmentation and classification, to complete the whole LP recognition task. However, since ALPR is a cascading system, the results of LP recognition depend largely on the accuracy of LP detection.

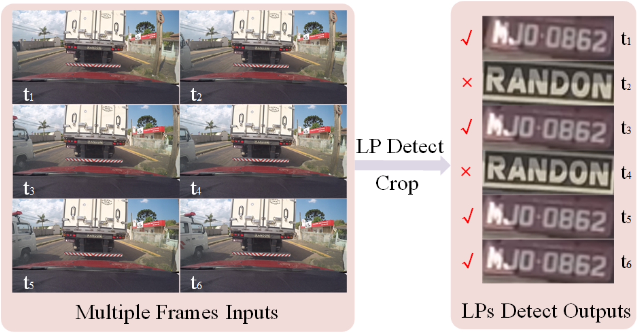

Figure 1: The results of the mainstream LPs detection methods in each image. What looks like LP may be misdetected, making the performance of the LPD systems worse.

With the advent of networks like YOLO [9, 10] and Faster R-CNN [11], the LP detection task has made a great breakthrough. In order to locate LPs efficiently, Li *et al.* utilize Faster R-CNN network to detect LPs in [7] and propose an end-to-end ALPR system. In this way, the direct detection of LPs in unconstrained scenarios may bring a lot of false positive (FP) samples. Based on this, the authors in [5, 6, 8] introduce a vehicle detection module before the LP detection stage to reduce the number of FP samples. However, these multi-stage approaches also increase the running time of the model.

It is worth noting that the above methods are all based on images for ALPR systems. However, many LP detection and recognition tasks in real scenes are carried out under surveillance videos, which provide more information than images. In [5], the authors release a multi-frame based LP recognition datasets named UFPR-ALPR. Laroca *et al.* [5] detect vehicles, LPs, and characters in turn and vote on multiple frames of results to determine the final LP recognition results. Nevertheless, this only makes use of some voting mechanisms to identify the characters and does not fuse contextual information at the feature level. As can be seen from Figure 1, some places like LPs on the vehicles may be misdetected, but the detection results of their adjacent frames do not go wrong.

As mentioned above, although the LPD systems have made a lot of progress, there are still many problems to be addressed. (1) Most of the existing algorithms are implemented under specific conditions and are difficult to be applied in unconstrained scenarios.

[*] Qi Wang is the corresponding author. This work was supported by the National Natural Science Foundation of China under Grant U1864204, 61773316, 61632018, and 61825603.

Conference'21, August 2021, Taipei, Taiwan
2021. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

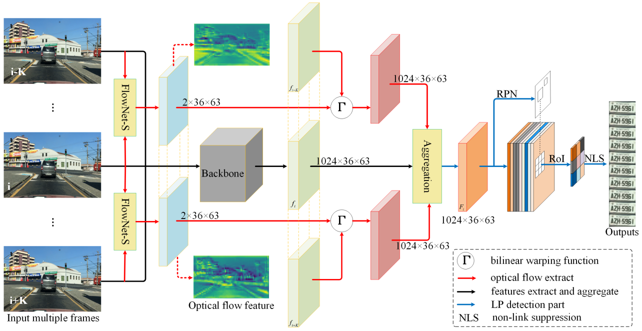

Figure 2: The whole framework of the AWFA-LPD network. There are two branches of the system, one is used to extract features of each input frame and the other is utilized to obtain optical flow feature maps between surrounding frames and reference frame I_i. The obtained features are then sent to the *Aggregation* module for adaptive weight feature fusion to obtain the new feature maps and complete the following detection tasks. Finally, *NLS* is used to post-process the detection results.

When the images are blurred, LPs are obscured, or the weather is rainy or foggy, they do not work well. (2) In order to achieve good results, many algorithms add multiple modules to enhance detection, which increase the complexity of the systems. (3) Most research today is based on images as input. However, in practical applications, there may be continuous multiple frames as input. If the context information of local frames can be well fused, the detection results in reference frame will be greatly improved..

To remedy these issues, we design an adaptive weight guided feature aggregation network named AWFA-LPD in this paper, which can well fuse the information of nearby frames and obtain a desirable effect. The whole network propagates the information of these frames through optical flow extraction. Then it allocates the weight of the feature maps obtained from the input multi-frame sequence through an adaptive weight calculation module, and finally aggregates them for subsequent detection task. Moreover, we also concatenate a non-link suppression mechanism at the end to establish the spatial relationship between the detection results of each frame, effectively suppressing FP samples similar to LPs. The main contributions of this article are as follows:

- An unified deep neural network combining optical flow extraction is proposed, which can fuse the feature maps of local frames to enhance the detection of the reference frame.
- A non-link suppression module is embedded in the system, which can be used for post-processing of the detection results.
- Compared with other image based LP detection methods, this LPD framework achieves state-of-the-art (SOTA) results.

2 PROPOSED LPD METHOD

In this section, we provide an overview of the entire LPD framework, and then explain the details of each implementation.

2.1 Overview

Generally speaking, a complete and efficient LPD system is able to accurately detect LPs in images or videos in unconstrained scenarios. As mentioned above, [5, 6, 8] concatenate the vehicle detection stage to reduce the FP samples in images, which will increase model complexity and cannot suppress the FP samples on the vehicles. In many practical applications, the inputs to the LPD systems can be a continuous multi-frame images or videos. In this case, we design a model to directly detect LPs in this paper, which fuses the feature maps of adjacent frames.

As can be seen from Figure 2, the proposed AWFA-LPD consists of three components: the optical flow alignment module, the adaptive feature aggregation module, and the non-link suppression module. The inputs of the entire network are video frames $\{I_i\}, i = 1, \dots, \infty$, and each frame will pass through the shared convolutional neural networks (CNNs) to get the feature maps $\{f_i\}$. The local feature maps will be propagated to the reference frame through the optical flow alignment module, and the weight w of these features will be calculated by the adaptive weight feature aggregation module and the new feature map $\{F_i\}$ of the reference frame will be obtained by aggregating these features. Finally, $\{F_i\}$ is sent to the detection network to output the bounding boxes of the reference frame's LPs. Then get the outputs of each frame in turn and the non-link suppression module is used to suppress the isolated LPs in the sequence and output the final detection results.

2.2 Optical Flow Alignment Module

In our proposed AWFA-LPD, we utilize ResNet-50 [4] as our backbone network to extract features from video frames. It is worth mentioning that the extracted feature maps $\{f_i\}$ are not spatially aligned due to the motion of the objects in videos. For example, if we directly fuse several feature maps of consecutive frames, LPs in frame $t-1$ will offset their location in frame t. In order to alleviate this problem, the authors in [17–19] apply an optical flow network in the field of video object detection to propagate the feature maps between frames. Inspired by this, we utilize an optical flow alignment module to correct the feature maps $\{f_i\}$ before aggregating them.

Since there is no optical flow groundtruth in the video based datasets, we utilize FlowNet-Simple [2] to extract optical flow features here, which has been pre-trained on the Flying Chairs dataset [2]. After obtaining the flow features, the feature maps f_j on the neighbor frame I_j are warped to the reference frame I_i by a warping function, given by

$$f_{j\to i} = \Gamma\left(f_j, N_{flow}\left(I_i, I_j\right)\right), \tag{1}$$

where Γ is a bilinear warping function applied to each channel in the feature maps, N_{flow} represents the optical flow extraction network and $f_{j\to i}$ denotes the feature maps warped from neighbor frame I_j to the reference frame I_i. It can be seen from Figure 2 that the visualization of optical flow features $N_{flow}\left(I_i, I_j\right)$ between frame j and frame i.

2.3 Adaptive Weight Feature Aggregation Module

The feature maps obtained after feature warping need to be aggregated and sent to the next detection network. In this part, we design a feature aggregation module with adaptive weight, which can assign different weight to the feature maps of adjacent frames and aggregate them with the feature maps f_i of the reference frame I_i into a new feature map F_i.

Adaptive Weight. As the frame interval is larger, the difference of the feature maps is larger. In this case, we expect feature maps that are very different from reference frame to have as little impact as possible. Here, we utilize the cosine similarity metric to calculate the similarity between two feature maps. Given a parameter K representing local frame specifications and the input feature maps buffer $[f_{i-K}, \ldots, f_{i+K}]$, if the local features f_j is close to the features f_i, it will output a large weight. Here we calculate the cosine similarity of the features to assign the weight. The weight estimation formula is given by

$$w_{j\to i} = exp\left(\frac{f_{j\to i} \cdot f_i}{\left|f_{j\to i}\right| \left|f_i\right|}\right), \tag{2}$$

where $w_{j\to i}$ is a normalized weight and $\sum_{j=i-K}^{i+K} w_{j\to i} = 1$. What's more, instead of getting the weight directly through softmax, we use the temperature parameter T [14] to control the weight distribution here and the softmax-T formula is given by

$$\bar{w}_{j\to i} = \frac{exp\left(w_{j\to i}/T\right)}{\sum_{j=i-K}^{i+K} exp\left(w_{j\to i}/T\right)}. \tag{3}$$

It can be seen that when T is larger, the weight is smoother, on the contrary it is sharper.

After optical flow alignment and weight calculation, the features F_i of the reference frame will be obtained by

$$F_i = \sum_{j=i-K}^{i+K} \bar{w}_{j\to i} f_{j\to i}, \tag{4}$$

and F_i are then sent to the detection network to locate the LPs in the reference frame.

2.4 Non-link Suppression Mechanism

F_i input to the detection network will output the bounding boxes (bboxes) and scores of LPs in the reference frame. Although the features of nearby frames have been aggregated in F_i, if the detection results of surrounding frames can be combined at box-level, better results will be achieved.

In [3], Han *et al.* propose the Seq-NMS to establish the connection between the output bboxes. However, Seq-NMS connects bounding boxes first and then carries out non-maximum suppression (NMS) operation, whose results depend largely on the connection results. As the LP is small, Seq-NMS will cause deviation to the overall detection results.

In this case, we propose the non-link suppression mechanism (NLS). NLS first uses NMS to get preliminary detection results in each frame, and then connects the bboxes of each frame. For example, when the Intersection over Union (IoU) of the bbox of I_i and the bbox of I_{i+1} is greater than 0.3, the bbox is considered to be the same LP and a connection is established. If there is a bbox that cannot connect to any bboxes of adjacent frames, remove it. The function of NLS is to suppress the number of isolated FP samples, effectively increasing precision.

2.5 Network Structure

As shown in Figure 2, we utilize the ResNet50 as the shared backbone network and R-FCN as the detection network.

Backbone. The input frames are resized to a size of 562 × 1000 before being sent to the network. For Resnet-50 [4], we discard the average pooling layer and the 1000-d fully convolutional layer, only using the convolutional layers to extract feature maps. At the same time, we also remove the conv5 block so that the final output is 1024 channels.

Optical Flow Network. As shown in Figure 2, the size of the features extracted from the shared convolutional neural networks is $1024 \times 36 \times 63$, which is $\frac{1}{16}$ of the input images. To ensure that the output of the optical flow extraction is the same size as the output of the backbone, we reduce the size by appending a pooling layer before feeding it into the optical flow network. To get the optical flow features between frames, we utilize the CNN based network named Flownet-Simple [2] here to extract the optical flow feature maps. As the existing datasets do not have the groundtruth of optical flow features, we directly load the model parameters for training on the Flying Chairs dataset [2].

Detection Network. Compared with Faster-RCNN, R-FCN has a faster detection speed and even higher accuracy. After obtaining the aggregated feature maps, there will be two branches of full convolutional networks, respectively used for region proposal and detection. In the region proposal network (RPN), two sibing 1×1 convolutional layers are appended to output object scores and bounding boxes, respectively. In the detection part, there are also two 1×1 convolutional layers output the position-sensitive score maps and bounding box regression maps, Their dimensions are k^2 $(C + 1)$ and $4k^2$, where C denotes the object categories (+1 for background) and k^2 denotes a $k \times k$ saptial grid describing relative positions. Here, we take $C = 1$, $k = 3$, then k^2 encodes the cases of *{top-left, top-center, top-right, ..., bottom-right}* of an object category. Finally, region of interset (ROI) align, voting by averaging the k^2 scores and non-maximum suppression (NMS) are used to obtain the classification scores and bboxes regression results of each frame.

3 EXPERIMENTS

In this section, we will describe the experimental details and present the final results.

3.1 Datasets and Implementation Settings

In order to verify that the LPD system is effective, we do experiments on the video based dataset named UFPR-ALPR [5]. This dataset contains 60 videos for training, 30 for validation and 60 for testing. Compared with the ordinary LP datasets, it has two characteristics, which are based on continuous multiple frames and complex LPs with motorcycles inside. Moreover, UFPR-ALPR has 30 frames in each video.

During the experiment, we set K to 2, that is, the total number of frames to be input as 5. If too many frames are input, the model will take too long to run. Moreover, we set T to 0.1 to give more confidence to the features that are closer to the feature maps of the reference frame. We make all experiments on a computer with an Intel Core 3.4GHz CPU, 12GB of RAM and four NVIDIA 1080Ti GPU.

3.2 Performance Evaluation and Ablation Study

We compare the results with YOLOv3 [10], EAST [16], R-FCN [1], FGFA [18], and the method in original dataset [5]. Since object category we detected is only LP, the detection accuracy is measured by precision and recall rate here, which makes the specific test results be more clear. In order to comprehensively consider precision and recall, we used F1-score as a supplementary evaluation method. The formula is as follows:

$$F1 - score = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{5}$$

The running time of the model can evaluate the efficiency of the model. Here, we calculate the time taken by the model in each algorithm to process the whole image as runtime. Referring to YOLO and Faster-RCNN, we only consider the IoU between the detection result and the groundtruth greater than or equal to 0.5 as true positive sample.

Table 1: LP detection recall and precision (percentage) of state-of-the-art detecion models on UFPR-ALPR.

Methods	Recall	Precision	F1-score	runtime(ms)
YOLOv3	97.39	94.91	96.13	21
EAST	99.89	92.78	96.2	26
R-FCN	99.83	94.56	94.56	66
Laroca *et al.*	98.33	-	-	**16**
FGFA	98.28	97.19	97.19	87
AWFA-LPD(Ours)	**100**	**97.30**	**98.63**	78

Table 2: Ablation studies on UFPR-ALPR. The optical flow extraction module and NLS module are removed respectively to verify the effectiveness of our method.

Methods	Recall	Precision	F1-score	runtime(ms)
AWFA(not flow+NLS)	98.44	95.39	96.89	**68**
AWFA(not flow)	98.44	96.41	97.41	73
AWFA(not NLS)	100	96.98	98.47	76
AWFA-LPD(Ours)	**100**	**97.30**	**98.63**	78

In order to study the impact of T and K on our system, we verify other values of K and T. When K is less than 5, set $K = 2$ to get the maximum recall rate and precision. When K is greater than 5, although the precision is 98.03% when $K = 7$, the runtime exceeds 150ms and is not efficient. Similarly, compared with $T = 0.01$, 0.5 and 1, the best effect is obtained when $T = 0.1$. The experimental results on UFPR-ALPR are shown in Table 1. Since Laroca *et al.* method only gives the recall result, we use – instead of precision here. It is not difficult to see that our method achieves satisfactory results in the accuracy of detection. Since our model is integrated with the optical flow extraction module, it does not have a great advantage in running time. In addition, the precision rate and recall rate are higher than any other algorithm with known results. The recall rate reaches 100% in the UFPR-ALPR dataset, that is, all positive samples are detected. In this aspect, state-of-the-art (SOTA) effect is achieved.

We next conduct ablation studies to analysis the impact of each component in our method. As can be seen from Table 2, when the optical flow extraction module is removed, although the running time is reduced, the recall rate and precison rate are both decreased. If we do not add NLS module as post-processing, the recall rate is not affected, but the precision rate is slightly reduced. It can be concluded that each component of our approach has a positive effect on the overall framework.

4 CONCLUSIONS

In this work, we propose a license plate detection architecture named AWFA-LPD based on deep neural networks, which integrates the features of local frames. AWFA-LPD involves an optical flow extraction network to align and propagate feature maps of adjacent frames and adaptive weight feature aggregation module to fuse features. Extensive comparison experiments prove the effectiveness and efficiency of the proposed LPD framework. Due to the introduction of optical flow module, there is a partial runtime loss. Our future work will focus on how to develop more lightweight feature fusion methods to replace optical flow.

REFERENCES

[1] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*. 379–387.

[2] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 2758–2766.

[3] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. 2016. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465* (2016).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[5] Rayson Laroca, Evair Severo, Luiz A Zanlorensi, Luiz S Oliveira, Gabriel Resende Gonçalves, William Robson Schwartz, and David Menotti. 2018. A robust real-time automatic license plate recognition based on the YOLO detector. In *2018 international joint conference on neural networks (ijcnn)*. IEEE, 1–10.

[6] Rayson Laroca, Luiz A Zanlorensi, Gabriel R Gonçalves, Eduardo Todt, William Robson Schwartz, and David Menotti. 2019. An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *arXiv preprint arXiv:1909.01754* (2019).

[7] Hui Li, Peng Wang, Mingyu You, and Chunhua Shen. 2018. Reading car license plates using deep neural networks. *Image and Vision Computing* 72 (2018), 14–23.

[8] Sergio Montazzolli Silva and Claudio Rosito Jung. 2018. License plate detection and recognition in unconstrained scenarios. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 580–596.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

[10] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[12] Sergio Montazzolli Silva and Claudio Rosito Jung. 2017. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*. IEEE, 55–62.

[13] Qi Wang, Junyu Gao, and Yuan Yuan. 2017. Embedding structured contour and location prior in siamesed fully convolutional networks for road detection. *IEEE Transactions on Intelligent Transportation Systems* 19, 1 (2017), 230–241.

[14] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.

[15] Yuan Yuan, Zhitong Xiong, and Qi Wang. 2016. An incremental framework for video-based traffic sign detection, tracking, and recognition. *IEEE Transactions on Intelligent Transportation Systems* 18, 7 (2016), 1918–1929.

[16] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 5551–5560.

[17] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. 2018. Towards high performance video object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7210–7218.

[18] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 408–417.

[19] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. 2017. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2349–2358.