FF-LPD: A Real-Time Frame-by-Frame License Plate Detector with Knowledge Distillation and Feature Propagation

Haoxuan Ding, Junyu Gao, *Member, IEEE,* Yuan Yuan, *Senior Member, IEEE,* and Qi Wang, *Senior Member, IEEE*

***Abstract*—With the increasing availability of cameras in vehicles, obtaining license plate (LP) information via on-board cameras has become feasible in traffic scenarios. LPs play a pivotal role in vehicle identification, making automatic LP detection (ALPD) a crucial area within traffic analysis. Recent advancements in deep learning have spurred a surge of studies in ALPD. However, the computational limitations of on-board devices hinder the performance of real-time ALPD systems for moving vehicles. Therefore, we propose a real-time frame-by-frame LP detector focusing on real-time accurate LP detection. Specifically, video frames are categorized into keyframes and non-keyframes. Keyframes are processed by a deeper network (high-level stream), while non-keyframes are handled by a lightweight network (low-level stream), significantly enhancing efficiency. To achieve accurate detection, we design a knowledge distillation strategy to boost the performance of low-level stream and a feature propagation method to introduce the temporal clues in video LP detection. Our contributions are: (1) A real-time frame-by-frame LP detector for video LP detection is proposed, achieving a competitive performance with popular one-stage LP detectors. (2) A simple feature-based knowledge distillation strategy is introduced to improve the low-level stream performance. (3) A spatial-temporal attention feature propagation method is designed to refine the features from non-keyframes guided by the memory features from keyframes, leveraging the inherent temporal correlation in videos. The ablation studies show the effectiveness of knowledge distillation strategy and feature propagation method.**

***Index Terms*—Automatic license plate detection, Knowledge distillation, Feature propagation, Spatial-temporal attention.**

I. Introduction

With the development of artificial intelligence and deep learning, the Automatic License Plate Detection (ALPD) systems [1]–[4] have been applied in traffic-related applications successfully. The well-known examples of ALPD applications are traffic law enforcement and parking lot access validation. Recently, most studies in ALPD have only focused on the LP detection for static vehicles. However, lots of scenarios in traffic and transportation contain moving vehicles, as illustrated in Fig. 1. And far too little attention has been paid to real-time video LP detection for moving vehicles. The LP detection for moving vehicles has a pivotal role in intelligent traffic issues. As for traffic law enforcement, there is an urgent need to address the LP detection and identification during tracking and chasing. In the field of intelligent transportation dispatching, real-time LP detection aids in identifying defective vehicles when faults occur and dispatching other vehicles to dodge the faulty ones. Therefore, real-time video LP detection in motion is necessary for traffic-related applications.

Fig. 1. Some movement scenarios in transportation.

The recent advances in deep learning have contributed to computer vision tasks, especially object detection, significantly. Convolutional Neural Networks (CNNs) have been introduced to the vehicle and LP detection [5]–[7], effectuating obvious advancement. Because of the motion, video LP detection task has a significant demand in real time. In order to accomplish the real-time detection, YOLO [8] and its varieties [9]–[11] are usually utilized in ALPD systems [12]–[14], but current real-time detection methods pursue the efficiency rather than accuracy, leading to low detection performance. Moreover, in order to achieve accurate detection, many detectors use deeper and larger neural networks, causing time-consuming inference. Thus, the main challenge faced by researchers is the trade-off between efficiency and accuracy in

This work was supported in part by the National Natural Science Foundation of China under Grant U21B2041.

Haoxuan Ding is with the Unmanned System Research Institute, and with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P. R. China. (e-mail: haoxuan.ding@mail.nwpu.edu.cn)

Junyu Gao, Yuan Yuan, and Qi Wang are with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China. (e-mail: gjy3035@gmail.com, y.yuan1.ieee@gmail.com, crabwq@gmail.com)

Qi Wang and Junyu Gao are the corresponding authors.

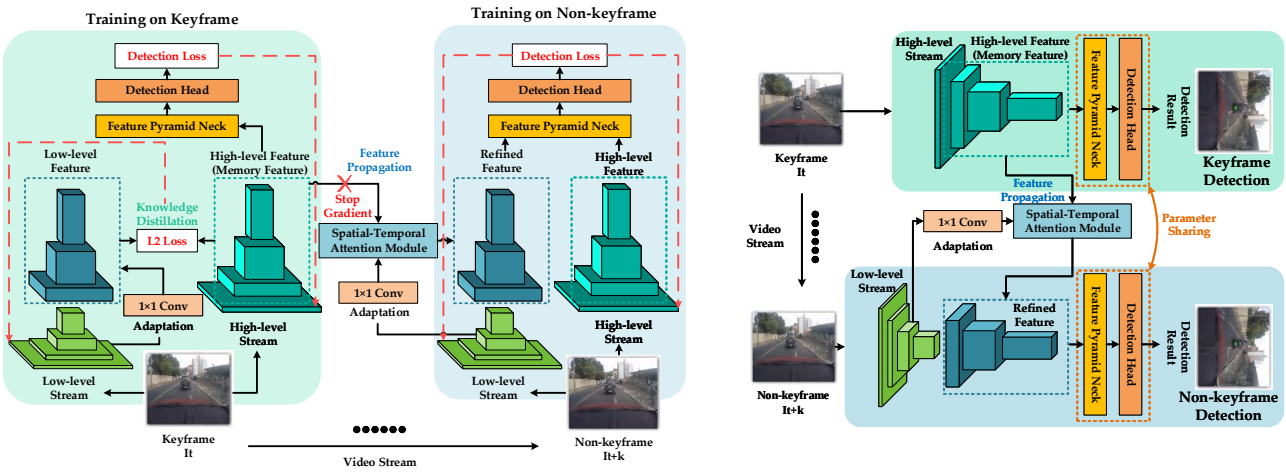


Fig. 2. Overview of the proposed FF-LPD framework. There are two streams in FF-LPD, high-level stream and low-level stream. The low-level stream mimic the high-level stream by the guidance of knowledge distillation. And the feature propagation method provides more extra cues for low-level stream detection. These two modules improve the low-level stream performance effectively. The red dotted lines in (a) denote the back propagation pathways. During inference, the high-level stream detects the LP on keyframes and the low-level stream detects the LP on non-keyframes.

video LP detection.

In order to procure real-time detection, a widespread way is to compress the deep network to a shallow network, reducing the parameters and accelerating the inference. Knowledge distillation is one of the prevalent network compression schemes in deep learning. It aims to utilize a pre-trained deeper teacher network to guide the training of a small student network. Under the instruction of the teacher network, the student network can acquire the performance close to the teacher network and accomplish faster inference. For the purpose of shrinking the runtime, we introduce the knowledge distillation strategy in training. In addition, video data have the inherent temporal correlation, and these extra correlation cues alleviate the deterioration in videos, such as motion blur and rare pose. To increase the reliability of detection, we design a feature propagation method to boost the video LP detection performance by linking the spatial-temporal correlation among frames. In a word, we contrive a knowledge distillation strategy and a feature propagation method in typical one-stage detector, proposing a real-time frame-by-frame video LP detector named FF-LPD.

In practice, the recent real-time LP detectors usually directly select the lightweight neural networks to improve the efficiency and there is no tricks in them, causing the poor convergency and low accuracy. In the proposed FF-LPD, the video frames are divided into keyframes and non-keyframes before processing. Then, a deep high-level stream (teacher network) is used to detect LPs from keyframes, and a shallow low-level stream (student network) is used to detect LPs from non-keyframes. For the accurate performance of the low-level stream, we design a simple feature-based knowledge distillation strategy to update the low-level stream with the guidance from the high-level stream in joint training.

Meanwhile, we fully exert the intrinsic correlation in video data to assist the non-keyframes detection, proposing two feature propagation methods based on 3D convolution or transformer respectively. Recently, the feature propagation methods usually base on optical flow and visual attention. However, the extraction of optical flow is time-consuming. And some attention mechanisms only consider the spatial relation rather than combined spatial-temporal relation. The proposed feature propagation approaches extracts the spatial-temporal attention and attempts to fine-tune the imperfect low-level features from non-keyframes with the exquisite high-level memory features from interrelated keyframes. The performance of these two feature propagation methods are compared and we analyze their advantages and disadvantages from experiments. In addition, the ablation studies demonstrate that the proposed knowledge distillation strategy and feature propagation method significantly improve the accuracy for non-keyframe detection.

In summary, the main contributions of this paper are:

1) Propose a real-time frame-by-frame LP detector named FF-LPD for video LP detection. Compared with current real-time LP detectors, the proposed FF-LPD gains the higher accuracy by introducing the frame-by-frame strategy. It tackles the keyframes and non-keyframes through two backbone streams, achieving a real-time (48.4 FPS) LP detection with a competitive performance compared to the state-of-the-art real-time one-stage detectors.
2) To further inspire the performance of the lightweight backbone, we introduce a simple feature-based knowledge distillation strategy to prompt the performance of low-level stream for non-keyframes detection, constraining the latent feature space of the low-level stream by the guidance from the high-level stream.
3) To propagate the intrinsic correlation efficiently, we attempt to build the spatial-temporal attention for feature propagation by 3D convolution and transformer. Com-

pared with current feature propagation methods, the 3D convolution based feature propagation method is light enough to avoid the soar of the number of parameters.

The rest of this paper is organized as follows. Section II reviews the related works. Section III describes the proposed FF-LPD in details. Section IV illustrates the experimental setting and results on video LP detection datasets and analyzes the effect of our proposed method. Eventually, we summarize the whole work in Section V.

II. Related Works

In this section, we briefly summarize the related deep learning approaches about the ALPD system in this paper. The LP detection is a sub-field of object detection, so that the object detection methods and LP detection methods are illustrated first. Then, the introduced methods in FF-LPD are exhibited, reviewing the recent knowledge distillation methods and feature propagation methods.

A. Object Detection

The deep learning methods, especially the convolutional neural network (CNN), have achieved significant improvement on computer vision tasks. Generally, the object detector is divided into two categories: one-stage and two-stage detectors. The core difference between these two detectors is how the region proposals are obtained. Two-stage detectors usually first obtain the region proposals in the first stage, and then extract the features from those region proposals for classification in the second stage. R-CNN [15] is one of the first two-stage detectors with CNN. Afterwards, the Fast R-CNN [16] improves the inference speed by extracting features only once, and a RoIPooling layer is proposed to map the image contents to feature map patches. Similarly, SPP-Net [17] also extracts features of region proposals with several scales pooling layers. To acquire more accurate region proposals, Faster R-CNN [18] presents a Region Proposal Network (RPN) to generate region proposals through CNN and introduces the anchors in object detection. To alleviate the misalign between features and objects, Mask R-CNN [19] introduces the RoIAlign layer in Faster R-CNN [16] to align the object areas between the image and feature. In order to improve the quality of region proposals, Cascade RPN [20] proposes adaptive convolution to align the anchors and features.

One-stage detectors start with YOLO [8]. YOLO is proposed to directly classify the object categories and regress the bounding box through a CNN model. Meanwhile, Single Shot MultiBox Detector (SSD) [21] is also a one-stage detector. Compared to YOLO [8], SSD [21] introduces anchor boxes and multi-scale features in classification and localization. Similarly, YOLOv2 [9] also introduces anchors in YOLO [8], achieving better detection performance. In practice, the utilization of multi-scale features plays a positive role in object detection, so that Feature Pyramid Neck (FPN) [22] proposes an inherent multi-scale pyramidal hierarchy of CNN to build feature pyramids, merging the features with different receptive fields. After that, the FPN is applied in object detection methods generally. YOLOv3 [10] merges different scale feature maps in YOLO [8]. To further improve the accuracy, YOLOv4 [11] introduces many tricks in YOLO [8]. Due to the elimination of region proposals, the one-stage detectors achieve quite high efficiency on object detection. The real-time detection methods, especially YOLO [8] and its variants, are applied in the industry detection tasks successfully. However, without the region proposals, the imbalance between foreground and background has an adverse impact on one-stage detectors. Thus, RetinaNet [23] proposes focal loss to handle the foreground and background imbalance issue.

Moreover, there are some methods break through the traditional detection methodology, innovating the object detection pipeline. For example, CenterNet [24], CornerNet [25] and CentripetalNet [26] detect objects through regressing and matching the keypoints on objects. And FCOS [27] first introduces the fully convolutional network (FCN) [28] into object detection.

In addition, the vision transformer is introduced to detection task recently, which builds up a new normal form for detection. DETR [29] is the first work to introduce transformer [30] into detection task. It provides a simple pipeline for object detection. DETR [29] considers the object detection as set prediction and predicts the categories and positions of objects. The success of DETR [29] creates the surge of detection transformers [31]–[33]. DeformDETR [34] introduce deformable attention mechanism to improve the model performance. Gao *et al.* [35] propose a Spatially Modulate Co-Attention (SMCA) to constrain the attention response to be high near initially estimated bounding box locations, accelerating the model convergence. CondDETR [36] learns a conditional spatial query for multi-head cross-attention in decoder. However, Wang *et al.* [37] find that detection transformers show superior performance only on datasets with rich training data like COCO 2017 [38], while the performance reduces significantly when the dataset is small. The reason why we do not attempt the transformer framework for this paper is because the video LP detetcion datasets are quite small and the detection transformer will suffer from overfitting problem.

B. LP Detection

LP detection is a sub-task in object detection. Many object detection methods mentioned above are applied in LP detection task. For example, there are some two-stage LP detection methods. Rafique *et al.* [5] apply Faster R-CNN [18] for LP detection. Dong *et al.* [6] also use a two-stage detector for LP detection and regress four corner points of LP. Li *et al.* [7] use RPN for LP detection and the RPN is trained for both detection and recognition.

As mentioned above, YOLO [8] and its variants are applied in the industry generally. Hsu *et al.* [12] utilize YOLO [8] and YOLOv2 [9] in LP detection task. Laroca *et al.* [13] use YOLO and Fast YOLO in the LP detection task and propose a large scale video LP detection dataset UFPR-ALPR. Furthermore, Laroca *et al.* [39] introduce layout classification into YOLO to detect LP from different countries, regions, and vehicles. Silva and Jung [14] also use Fast-YOLO to detect the frontal view and LP of cars. Gonçalves *et al.* [40] propose a simple CNN for LP detetion and a dataset named SSIG-ALPR.

In addition, Li and Shen [41] train a CNN through cropped character to achieve character-based LP detection. Zhang *et al.* [42] propose an optical-flow guided unified framework for license plate detection, tracking, and recognition. As for wild LP detection, Silva and Jung [43] propose the WPOD-NET to detect LP in unconstrained scenarios. Silva *et al.* [44] also propose a LP detector for unconstrained scenarios.

C. Knowledge Distillation

Knowledge distillation is a common network compression approach. It improves the performance of a lightweight student network through the guidance of a pre-trained teacher network with deeper architecture. Hinton *et al.* [45] propose the conception of knowledge distillation and the first knowledge distillation method. Zhang *et al.* [46] attempt to utilize the mutual knowledge in several student networks rather than the teacher network. Cho *et al.* [47] find that the capacity mismatching between teacher and student networks will cause the failure of distillation, and they propose an early-stop teacher regularization strategy to alleviate the effects of capacity mismatching. Meanwhile, Mirzadeh *et al.* [48] propose the teacher assistant network to alleviate the capacity mismatching. In order to mimic the teacher network better, Jin *et al.* [49] constrain the optimization route in student network training with checkpoints from the teacher network.

The knowledge distillation methods introduced above only focus on the similarity between outputs of the teacher and student networks. In addition, some methods concentrate on the features and relations among data. FitNets [50] is the first work that constrains the feature maps from hidden layers in the student network. Zagoruyko *et al.* [51] distills the knowledge through making the attention maps from the teacher and student similar. Yim *et al.* [52] learn the relations from different channels among hidden layers. Tung and Mori [53] find that the samples from the same category have similar activation results, and they propose the pairwise similarities to constrain the student with the guidance from teacher. Shin *et al.* [54] utilize an attention similarity knowledge distillation approach to boost low-resolution face recognition performance. Chen*et al.* [55] is the first method which introduced the knowledge distillation into object detection.

D. Feature Propagation

The feature propagation method is commonly used in frame-by-frame detectors. DFF [56] is the first method to propagate the keyframe feature to the non-keyframe feature using a flow-warping method based on optical flow. Meanwhile, FGFA [57] also introduces the optical flow to guide the feature aggregation. However, optical flow extraction is time-consuming, so that the methods based on optical flow are unsuitable for real-time detection. The attention mechanisms are also widely applied in feature propagation. Self-Attention [58] and Non-Local [59] are proposed to model the inherent correlation in data. MatchTrans [60] introduces Non-Local [59] in video object detection to propagate the features across frames. Shvets *et al.* [61] introduce the temporal relation module to propagate the support features to target features. MANet [62] combines the features from nearby frames to promote the robustness under occlusion in video object detection. MEGA [63] introduces the attention mechanism in Relation Net [64] to propagate the long-range memory features. With the development of vision transformer, some feature propagation methods utilize the self-attention and multi-head attention in transformer. For example, Chen *et al.* [65] design a feature fusion network based on transformer to correlate the objects in tracking task.

III. Approach

In this section, we first explain the overview of the proposed FF-LPD framework. Then, we show the knowledge distillation strategy in our approach. Next, the architecture of the spatial-temporal attention module for feature propagation is explained. Finally, the details of the training of FF-LPD are illustrated.

A. Framework Overview

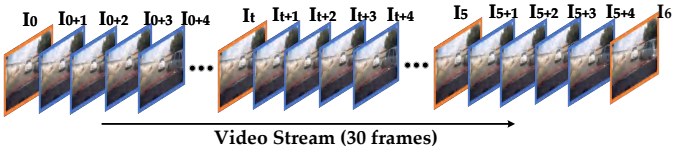


Fig. 3. The division of video frames. The orange edge frames are keyframes I_t, and the blue edge frames are non-keyframes I_{t+k}. The corresponding keyframe for non-keyframes I_{t+k} is the keyframe I_t.

The frame division is shown in Fig. 3. From the initial frame of the video with l frames, the keyframes are sampled every N steps ($N = 5$ in this paper), marked as $I_t, t = 0 \cdots \lfloor l/N \rfloor$, and the rest of frames are all non-keyframes, noted as I_{t+k}, where t is the corresponding keyframe index and $k = 1 \cdots N-1$ is the index interval. The choice of the hyper-parameter N has impact on the trade-off between detection accuracy and inference efficiency, and we discuss the influence of N in Section IV-E.

The proposed FF-LPD is shown in Fig. 2. Two streams, high-level stream (ResNet-50 backbone) and low-level stream (ResNet-18 backbone), are originated in the detector. The reason why we choose ResNet-50 and ResNet-18 as backbones is because the knowledge distillation and feature propagation in proposed FF-LPD all need homologous high-level and low-level features. The ResNet-50 for high-level stream and the ResNet-18 for low-level stream have similar architectures and the same spatial feature map dimension for feature-based knowledge distillation and visual clues propagation in feature propagation. Meanwhile, the detector in FF-LPD is based on the FCOS detector [27] which choose the ResNet [66] as backbone, and ResNet [66] has been frequently-used backbone in detection task. Under this condition, we finally choose ResNet-50 and ResNet-18 as backbones. The high-level stream extracts elaborate high-level features from keyframes, and the low-level stream obtains rough low-level features from non-keyframes. The high-level stream and low-level stream share a common FPN and detection head for prediction.

*1) **Training on Keyframes**:* In training on keyframes, the high-level stream extracts the multi-scale high-level feature C_i^{high} from the input keyframe I_t. After that, the high-level feature C_i^{high} is passed to the common FPN and detection head to measure the detection loss and train the high-level stream. Meanwhile, the low-level stream extracts the low-level feature C_i^{low} from I_t, and the low-level representation learning is reined by the high-level feature C_i^{high} through the proposed knowledge distillation strategy, guiding the low-level stream to mimic the high-level stream. On the keyframe training stage, the low-level stream is only trained by knowledge distillation loss. Moreover, the high-level feature C_i^{high} from keyframe I_t is maintained as memory feature C_i^{mem} to be used in the feature propagation method.

*2) **Training on Non-keyframes**:* The I_{t+k}, where $k = 1 \cdots N-1$ is defined as the non-keyframe correspond to keyframe I_t with k intervals, as demonstrated in Fig. 3. Once the unsophisticated low-level feature C_i^{low} is extracted from I_{t+k} by low-level stream, the memory feature C_i^{mem} from corresponding keyframe I_t is propagated and embedded into C_i^{low} through the spatial-temporal attention, acquiring the refined low-level feature C_i^{ref}. This feature propagation introduces the intrinsic correlation among frames and refines the learned low-level representation, benefiting the non-keyframes detection. Ultimately, C_i^{ref} is passed to FPN and detection head to measure the loss and train the low-level stream. Besides, in the non-keyframes training stage, the high-level stream is trained through detection loss independently.

*3) **Inference**:* During the inference, the keyframe I_t is passed to the high-level stream to gain high-level feature C_i^{high}. On obtaining the feature C_i^{high}, the FPN and detection head predict the bounding boxes of LP from C_i^{high}. In the meantime, C_i^{high} from I_t is maintained as memory feature C_i^{mem}. Once the non-keyframe I_{t+k} is fed to the low-level stream, the memory feature C_i^{mem} is propagated to the low-level stream and embedded into the low-level feature C_i^{low}, attaining the refined feature C_i^{ref}. And then, the C_i^{ref} is used to predict the LP bounding boxes on non-keyframes. The FF-LPD achieves real-time inference because the majority of video frames (*e.g.* non-keyframes) are detected by the lightweight low-level stream. In addition, our introduced knowledge distillation strategy and feature propagation method realize a significant advancement on the low-level stream detection accuracy.

B. Knowledge Distillation

After collecting the features C_i^{high}, C_i^{low} from keyframe I_t, the proposed feature-based knowledge distillation strategy inhibits the C_i^{low} from C_i^{high}. The knowledge distillation strategy is shown in Fig. 4.

The multi-scale features C_i^{high} and C_i^{low} embrace three different-scale feature maps from the outputs of $Conv3\_x$, $Conv4\_x$, and $Conv5\_x$ layers in ResNet-50 and ResNet-18 [66] respectively.

A knowledge distillation loss L_{KD} is introduced to ensure that the low-level stream can imitate the high-level stream and educe more discriminating representations. According to

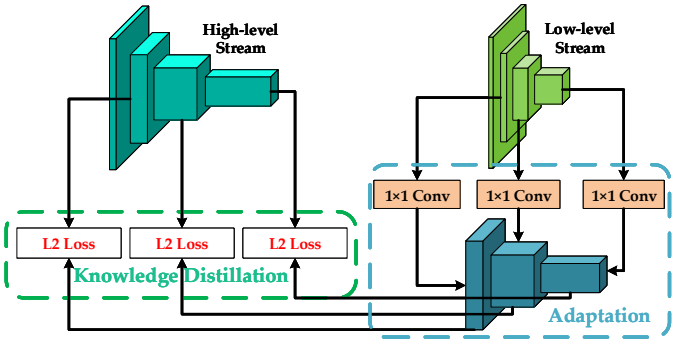


Fig. 4. The feature-based knowledge distillation strategy.

current feature-based knowledge distillation methods [50], [55], an adaptation module with 1×1 convolutional layer is used to align the channels between C_i^{high} and C_i^{low}, relieving feature discrepancy in distillation. The knowledge distillation loss L_{KD} is defined by Eq. 1:

$$L_{KD} = \sum_{i=3}^{5} \|C_i^{high} - C_i^{low}\|_2^2 \tag{1}$$

The dissimilarity between C_i^{low} and C_i^{high} is measured by $L2$ loss, referring to recent feature-based knowledge distillation methods [50]–[53]. The minimization of L_{KD} guides the low-level stream to mimic the high-level stream, gaining more discerning low-level features. On the basis of [46], [49], we train the teacher (high-level stream) and student (low-level stream) simultaneously rather than pre-training the teacher beforehand. In training on keyframes, the high-level stream is trained by detection loss, then we fix the high-level stream and train the low-level stream through knowledge distillation loss L_{KD}. The joint training guarantees that the update of the student can follow the teacher's optimization route.

C. Feature Propagation

The non-keyframe I_{t+k} and its corresponding keyframe I_t have spatial and temporal correlation, and the object detection benefits from these inherent correlation clues. In order to sufficiently implement this property in video data and elevate the LP detection accuracy, the memory feature C_i^{mem} from keyframe I_t is propagated to the low-level stream for detection on non-keyframe I_{t+k}. The memory feature C_i^{mem} is embedded into low-level feature C_i^{low} from I_{t+k}. To better propagate the temporal clues in video LP detection, we design two spatial-temporal attention for feature propagation, *i.e.* 3D convolution based and transformer based methods.

*1) **3D convolution based method**:* The 3D convolution based spatial-temporal attention module is demonstrated in Fig. 5. Our proposed 3D convolution based attention module elicits the spatial attention maps first, and then the 3D convolutional layer is introduced to link the temporal correlation. The details of attention method is illustrated in Algorithm 1. First, C_i^{mem} and C_i^{low} with three dimensions ($C \times H \times W$) are extended into four dimensions ($T \times C \times H \times W$), where T represents the temporal dimension. Then, the extended C_i^{mem} and C_i^{low} are concatenated together along temporal

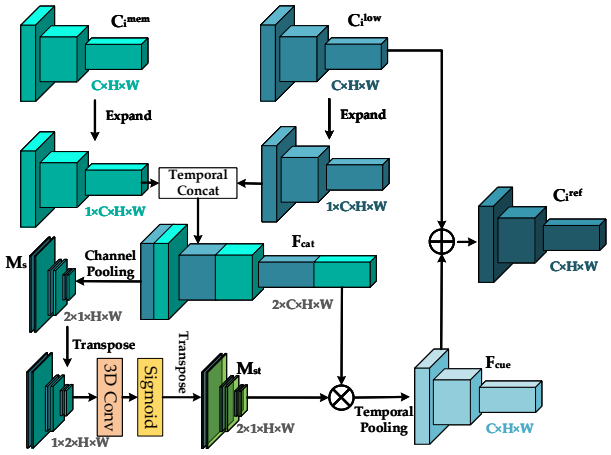


Fig. 5. 3D convolution based spatial-temporal attention mechanism for feature propagation. To clear illustrate the procedure, the shapes are signed nearby the corresponding feature maps.

Algorithm 1 The 3D convolution based spatial-temporal attention for feature propagation.

Require: Input memory feature C_i^{mem} from adjacent keyframe I_t and low-level feature C_i^{low} from current non-keyframe I_{t+k}.

Output: Refined feature C_i^{ref}.

1: Expand the input feature to $T \times C \times H \times W$
2: Concatenate the expanded feature along temporal dimension: $F_{cat} \leftarrow Concat(C_i^{mem}, C_i^{low})$
3: Channel pooling: $M_s \leftarrow Pooling(F_{cat})$
4: Transpose M_s to $C \times T \times H \times W$
5: Temporal link: $M_{st} \leftarrow Sigmoid(Conv_{3D}(M_s))$
6: Filter through attention: $F_{cue} \leftarrow M_{st} \odot F_{cat}$
7: Temporal pooling: $F_{cue} \leftarrow Pooling(F_{cue})$
8: Build skip connection: $C_i^{ref} \leftarrow F_{cue} \oplus C_i^{low}$
9: **return** C_i^{ref}

dimension, denoted as F_{cat}. After acquiring the F_{cat}, the spatial attention map M_s are obtained from F_{cat} through the global average pooling along channel dimension. Following this treatment, the 3D convolutional layer tackles the M_s and learns the spatial-temporal correlation between I_t and I_{t+k}, getting the spatial-temporal attention map M_{st}. The reason why we apply the 3D convolution on feature after pooling (*i.e.* M_s) rather than C_i^{mem} and C_i^{low} is because we need to reduce the computation consumption as much as possible to ensure the detection efficiency. M_s only has 1 channel after pooling, and the channel depth of C_i^{mem} and C_i^{low} are far more than M_s, leading to extremely increase on consumption. Under this condition, we finally decide use the 3D convolution on M_s instead of features or frames. On obtaining M_{st}, F_{cat} is filtered by spatial-temporal attention map M_{st}, gaining the cue feature F_{cue} with intrinsic spatial-temporal correlation. In order to adapting the dimension, the global average pooling along temporal dimension is implemented to squeeze the F_{cue} to the same shape as C_i^{low}. The final stage of this attention module is to build the skip connection between F_{cue} and C_i^{low} with addition, receiving the refined feature C_i^{ref}.

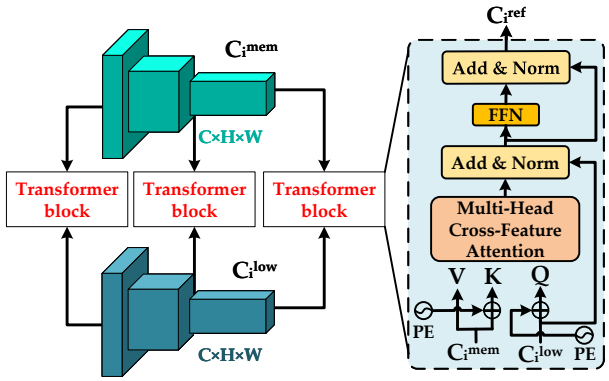


Fig. 6. Transformer based spatial-temporal attention mechanism for feature propagation.

*2) **Transformer based method**:* The transformer based spatial-temporal attention module is demonstrated in Fig. 6. The proposed transformer based attention module first generate the query features Q, key features K, and value features V, which is shown in Eq. 2:

$$\begin{aligned} Q &= C_i^{low} + PE, \\ K &= C_i^{mem} + PE, \\ V &= C_i^{mem}, \end{aligned} \tag{2}$$

where PE is the 2D absolute sine-cosine version position embedding on spatial dimension which is according to MAE [67].

The Q, K, and V with the size of $C \times H \times W$ are flattened into vector sequences S_Q, S_K, and S_V with the size of $C \times L$ where $L = H \times W$ is the sequence length. The S_Q, S_K, and S_V are input to a multi-head cross-feature attention layer to model the spatial-temporal relation between non-keyframe and keyframe and propagate visual clues. The multi-head cross-feature attention is defined as Eq. 3:

$$\begin{aligned} &MultiHead(S_Q, S_K, S_V) = Concat(H_1, ..., H_n)W^O, \\ &H_i = softmax(\frac{S_Q W_i^Q (S_K W_i^K)^T}{\sqrt{d_k}}) S_V W_i^V, \end{aligned} \tag{3}$$

where S_Q, S_K, S_V are the query, key, and value vector sequences. n is the number of attention heads, and we employ $n = 8$ in this work according to [65]. The d_k is the key vector dimensionality. The W_i^Q, W_i^K, W_i^V, and W^O are learnable parameters for multi-head attention.

After that, the residual skip connection and layer normalization are applied to the output feature from multi-head cross-feature attention layer. Then, a fully connected feed-forward network (FFN) with two linear layers and ReLU activation is used to process the features to enhance the fitting ability. Finally, the outputs of FFN are added with the inputs of FFN and normalized by layer normalization. Thus, the mechanism of transformer based method is summarize as:

$$\begin{aligned} S_i^{ref} &= LN(X + FFN(X)), \\ X &= LN(S_Q + MultiHead(S_Q, S_K, S_V)), \end{aligned} \tag{4}$$

where $LN(\cdot)$ is the layer normalization. $FFN(\cdot)$ is the feed-forward network. S_Q, S_K, S_V are defined in Eq. 2. And

$MultiHead(\cdot)$ is illustrated in Eq. 3. The output of FFN S_i^{ref} is reshape to the refined feature map C_i^{ref} with the size of $C \times H \times W$ for detection.

The above transformer based method build the correlation between non-keyframes and keyframes. The corss-feature multi-head attention guides the low-level stream to gain the visual clues from high-level stream for accurate detection.

D. Other Details

To keep the efficiency, the proposed FF-LPD is a one-stage anchor-free detection method, and we directly use the parameter-sharing FPN [22] and detection head in the current high-performance detection method FCOS [27].

*1) **Feature Pyramid Neck (FPN)**:* The FPN merges the features with different down-sample strides to suit the objects with inconsistent scales. The input multi-scale feature C_i^{high} or C_i^{ref} is combined with the details in [22], receiving feature maps from five levels $\{P_3, P_4, P_5, P_6, P_7\}$, and these feature levels have strides 8, 16, 32, 64 and 128, respectively.

*2) **Detection Head**:* The conventional detection head has a classification branch to classify the object categories and a regression branch to regress the coordinates of objects. Moreover, the detection head of FCOS [27] introduces an extra branch to regress the object centerness defined by:

$$centerness^* = \sqrt{\frac{min(l^*, r^*)}{max(l^*, r^*)} \times \frac{min(t^*, b^*)}{max(t^*, b^*)}}, \tag{5}$$

where l^*, r^*, t^*, b^* are the distances from the location center to the four sides of the bounding box. The centerness effectively alleviates the low-quality predicted bounding boxes that are far away from the object center.

*3) **Loss Function**:* Detection loss function L is implemented to train the high-level stream on both the keyframes and non-keyframes. It includes classification loss L_{cls}, centerness loss L_{cnt}, and regression loss L_{reg}:

$$\begin{aligned} L_{Det} = & \frac{1}{N_{pos}} \sum_{x,y} L_{cls}(p_{x,y}^{cls}, c_{x,y}^{cls}) \\ & + \frac{1}{N_{pos}} \sum_{x,y} L_{cnt}(p_{x,y}^{cnt}, c_{x,y}^{cnt}) \\ & + \frac{1}{N_{pos}} \mathbb{1}_{\{c_{x,y}^{cls} > 0\}} L_{reg}(t_{x,y}, t_{x,y}^{gt}), \end{aligned} \tag{6}$$

where L_{cls}, L_{cnt}, and L_{reg} are the focal loss in [23], cross-entropy loss, and GIoU loss in [68] respectively. The $p_{x,y}^{cls}$ and $c_{x,y}^{cls}$ denote the predicted category and ground-truth category on the object x with category y. The $p_{x,y}^{cnt}$ and $c_{x,y}^{cnt}$ are the predicted centerness and ground-truth centerness. And the $t_{x,y}$ and $t_{x,y}^{gt}$ respectively represent the predicted and ground-truth localization coordinates.

The loss function $\mathcal{L}_{low}$ in Eq. 7 is used to train the low-level stream and the loss function $\mathcal{L}_{high}$ in Eq. 7 is the loss function of high-level stream.

$$\begin{aligned} \mathcal{L}_{low} &= \begin{cases} L_{Det} + L_{KD} & \text{Input=keyframe} \\ L_{Det} & \text{Input=non-keyframe} \end{cases}, \\ \mathcal{L}_{high} &= L_{Det}. \end{aligned} \tag{7}$$

*4) **Distillation and Propagation**:* The distillation translates the knowledge of keyframe from high-level stream to low-level stream. And the feature propagation translates the clues from keyframe to non-keyframe. The reason why we do not combine the knowledge distillation and feature propagation into an attention-based distillation strategy [51], [54] is because the in distillation the inputs of high-level and low-level streams are all from keyframe which has no shift in feature. But the feature propagation processing has the feature shift because the module propagates the clues from keyframe to non-keyframe. Thus, we decouple these two modules to avoid the confusion from feature shift in learning.

IV. Experiments

In this section, we show the comparative results of our proposed approach. Section IV-A introduces the evaluation metrics in video LP detection task. Section IV-B briefly describes the selected video LP detection datasets. Section IV-C lists the implementation details in the experiments. Section IV-D displays our LP detection performance on two selected datasets. Then, the ablation studies are discussed in Section IV-E. Section IV-F analyzes the probable causes of false predictions.

A. Evaluation Metrics

For evaluating the algorithm performance, similar to the general evaluation metrics in object detection, we adopt the following criteria:

$$Precision = \frac{TP}{TP + FP}, \tag{8}$$

$$Recall = \frac{TP}{TP + FN}, \tag{9}$$

$$IoU(bbox, gt) = \frac{bbox \cap gt}{bbox \cup gt}, \tag{10}$$

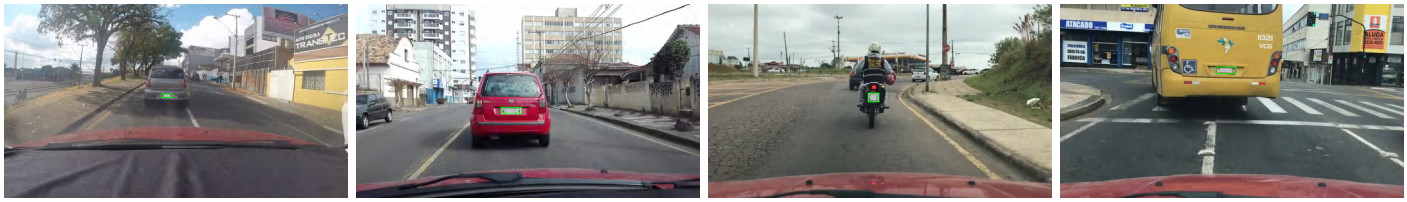
where TP, FP, and FN in Eq. 8 and Eq. 9 is the number of True Positive, False Positive, and False Negative samples under an IoU threshold τ ($\tau = 0.5$ in this paper). The $bbox$ and gt in Eq. 10 denote the predicted bounding box and corresponding ground-truth respectively. When the IoU (in Eq. 10) is greater than τ, the predicted result is considered as a True Positive sample. Otherwise, when the IoU is lower than τ, the predicted result is a False Positive sample. Besides, the omitted objects are treated as False Negative samples.

For the purpose of considerate evaluation, the evaluation metric should consider both the Precision and Recall simultaneously, so that a comprehensive metric Average Precision (AP) is generally used:

$$AP = \int_0^1 p(r) dr, \tag{11}$$

where $p(r)$ is the P-R curve. A higher AP means better performance. In addition, the average IoU is calculated by Eq. 12 among all predicted bounding boxes:

$$Average\text{–}IoU = \frac{1}{N_{bbox}} \sum_{i=1}^{N_{bbox}} IoU(bbox_i, gt_i), \tag{12}$$



(a) UFPR-ALPR Dataset

(b) SSIG-SegPlate Dataset

Fig. 7. Sample images with annotations in two selected datasets. The **green boxes** are annotations of LPs. (The LPs are covered by patches due to privacy constraints).

TABLE I
COMPARISON OF DIFFERENT APPROACHES ON UFPR-ALPR DATASET.

Method	Type	Backbone	AP	Average-IoU	FPS
Faster-RCNN [18]	two-stage	ResNet-50+FPN	94.60	0.7909	**27.9**
Cascade-RPN [20]		ResNet-50+FPN	**97.90**	**0.8367**	21.5
RetinaNet [23]	one-stage	ResNet-50+FPN	93.90	**0.7202**	29.6
YOLO-v3 [10]		DarkNet-53	90.20	0.7159	38.8
YOLO-v5 [70]		YOLOv5CSPDarknet	90.30	-*	31.0
YOLO-v7 [71]		YOLOv7Backbone	90.90	-*	36.0
YOLO-v8 [72]		YOLOv8CSPDarknet	95.10	-*	37.5
YOLOF [73]		ResNet-50	86.20	-*	40.0
YOLOX [74]		YOLOXCSPDarknet	94.50	-*	**41.8**
CentripetalNet [26]		Hourglass-104	**96.90**	0.7183	0.5
DFF [56]	VID	ResNet-50+FPN	92.30	-*	26.4
FGFA [57]		ResNet-50	91.82	-*	2.8
SELSA [75]		ResNet-50	93.23	-*	5.6
Temporal RoI Align [76]		ResNet-50	92.24	-*	1.8
FF-LPD-*3DConv* (ours)		ResNet-50/ResNet-18+FPN	94.98	**0.7610**	**48.4**
FF-LPD-*Trans* (ours)		ResNet-50/ResNet-18+FPN	**95.10**	0.7340	15.9

The methods except ours are reimplemented through open source object detection toolbox MMdetection [77], MMtracking [78], and MMYOLO [79]. All of methods are tested on a Nvidia GTX 1080TI GPU.
*The MMtracking [78] and MMYOLO [79] do not provide the statistics results about IoU.

where N_{bbox} denotes the number of all predicted bounding boxes. The Average-IoU manifests the LP localization accuracy.

B. Datasets

We select the video LP detection datasets UFPR-ALPR [13] and SSIG-SegPlate [69] to evaluate the proposed FF-LPD.

*1) **UFPR-ALPR Dataset**:* UFPR-ALPR dataset includes 150 annotated videos (4,500 images) with 150 vehicles in real-world scenarios, where both the cameras and tracked vehicles are moving. Every video has 30 frames encompassing only one plate. The video frames all have a fixed size of 1920×1080 pixels. In addition, the UFPR-ALPR dataset contains 120 videos of cars. The plates have the following types: gray car LP (90 videos), red car LP (30 videos), and motorcycles LP (30 videos). The dataset is split as follows: 40% for training, 40% for testing, and 20% for validation. Some images from the UFPR-ALPR dataset are shown in Fig. 7a.

*2) **SSIG-SegPlate Dataset**:* The SSIG-SegPlate dataset contains 101 on-track vehicle videos collected by a fixed camera (2000 images with the size of 1920×1080 pixels). The videos in the SSIG-SegPlate dataset have different video lengths. In addition, this dataset also embraces multiple vehicle types: passenger vehicles (1762 frames), buses and trucks (118 frames), others (120 frames). The SSIG-SegPlate has the same splits as UFPR-ALPR. Some samples from the SSIG-SegPlate dataset are demonstrated in Fig. 7b.

C. Implementation Details and Experimental Setting

Prior to the training or inference, original frames are resized to a fixed size of 512×512. To begin the training, we fix the learning rate at 10^{-3} during the first 10 epochs. Following this process, the learning rate decreases 10 times every 5 epochs. The momentum and weight decay are 0.9 and 0.0001, respectively.

TABLE II
COMPARISON OF DIFFERENT APPROACHES ON SSIG-SEGPLATE DATASET.

Method	Type	Backbone	AP	Average-IoU	FPS
Faster-RCNN [18]	two-stage	ResNet-50+FPN	**97.40**	0.5848	**27.9**
Cascade-RPN [20]		ResNet-50+FPN	96.60	**0.8239**	21.5
RetinaNet [23]	one-stage	ResNet-50+FPN	**98.60**	0.5949	29.6
YOLO-v3 [10]		DarkNet-53	93.50	0.6670	38.8
YOLO-v5 [70]		YOLOv5CSPDarknet	95.00	-*	31.0
YOLO-v7 [71]		YOLOv7Backbone	96.40	-*	36.0
YOLO-v8 [72]		YOLOv8CSPDarknet	97.10	-*	37.5
YOLOF [73]		ResNet-50	95.20	-*	40.0
YOLOX [74]		YOLOXCSPDarknet	98.20	-*	**41.8**
CentripetalNet [26]		Hourglass-104	98.10	**0.6873**	0.5
DFF [56]	VID	ResNet-50+FPN	96.34	-*	26.4
FGFA [57]		ResNet-50	94.23	-*	2.8
SELSA [75]		ResNet-50	97.61	-*	5.6
Temporal RoI Align [76]		ResNet-50	97.45	-*	1.8
FF-LPD-*3DConv* (ours)		ResNet-50/ResNet-18+FPN	97.69	**0.7979**	**48.4**
FF-LPD-*Trans* (ours)		ResNet-50/ResNet-18+FPN	**98.53**	0.7583	15.9

The methods except ours are reimplemented through open source object detection toolbox MMdetection [77], MMtracking [78], and MMYOLO [79]. All of methods are tested on a Nvidia GTX 1080TI GPU.
*The MMtracking [78] and MMYOLO [79] do not provide the statistics results about IoU.

Fig. 8. The comparison of detection performance among video detectors on some hard samples form UFPR-ALPR dataset [13]. The local areas of predictions are zoomed in to demonstrate the details of predictions from detectors. The false predictions have the red edging and red cross.

The metrics, AP and Average-IoU, are gauged on both the UFPR-ALPR dataset and the SSIG-SegPlate dataset. Meanwhile, in order to analyze the efficiency of our model, some metrics to measure the time consumption are gauged, consisting of FPS, GFLOPs, and the number of parameters.

The experimental environment is equipped with Intel(R) CPU Core(TM) i7-6900K @ 3.4GHz, 64GB RAM, and a NVIDIA GTX 1080TI GPU. As for the software environment, we employ the Pytorch [80] framework to build our model.

D. Performance on LP Detection

We compare the proposed FF-LPD with other state-of-the-art detectors on UFPR-ALPR and SSIG-SegPlate datasets. And both the two-stage, one-stage detectors, and video detector (VID) are selected in comparison. The comparison results on the UFPR-ALPR dataset [13] are shown in Table I. As for two-stage detectors, the Cascade-RPN [20] reaches the best detection accuracy (97.90 AP) among the listed models. However, compared with one-stage detectors, the two-stage Cascade-RPN [20] is mostly restricted to non-real-time LP detection due to the lack of efficiency (21.5 FPS). Likewise, although the CentripetalNet [26] achieves the top detection accuracy (96.90 AP) in compared one-stage detectors, the inference of CentripetalNet [26] is time-consuming (0.5 FPS) because of the large backbone Hourglass-104 [81], which means the CentripetalNet [26] is also unsuitable for real-time detection. The input images are with the size of 512×512 pixels, which limits the performance of YOLO-based methods [10], [70]–[74]. The LPs in images are quite small (smaller than 16×16 pixels) which is not suitable for YOLO-based

methods due to the limited resolution and unclear features of the small objects [82]. And compared to YOLO-based methods, our proposed FF-LPD-*3DConv* has higher operation efficiency. Compared with the video LP detection methods, our proposed FF-LPD-*3DConv* and FF-LPD-*Trans* achieve the better detection performance compared to other video detection methods. The FF-LPD-*Trans* reaches the best detection accuracy (95.10 AP) in video methods, but the efficiency of FF-LPD-*Trans* is quite low (15.9 FPS) which is unsuitable for traffic application. Meanwhile, FF-LPD-*3DConv* has the second place (94.98 AP) on detection performance among video methods, and proposed FF-LPD-*3DConv* reaches the highest efficiency (48.4 FPS) among all compared methods. As for video object detectors, the DFF [56] utilizes the FPN to fuse multi-scale features which is beneficial to the detection of small objects such as LPs. Thus, the detection accuracy of DFF [56] is higher than FGFA [57], which means the utilization of FPN can improve the detection performance. Therefore, we use the FPN architecture in the proposed FF-LPD. The efficiency of feature aggregation methods in FGFA [57], SELSA [75], and Temporal RoI Align [76] are time-consuming (lower than 10 FPS) and these video detectors are not suitable for LP detection in traffic application. Compared with them, our proposed feature propagation method (3D convolutional based or transformer based) has lower computing consumption and faster inference speed in temporal relation modeling. The proposed knowledge distillation module improves the detection accuracy of low-level stream dramatically. These two modules make the proposed FF-LPD exceeds the compared video detectors on both efficiency and accuracy. In summary, the proposed FF-LPD-*3DConv* and FF-LPD-*Trans* follow the Cascade-RPN [20] and CentripetalNet [26] closely, achieving the competitive detection performance compared to recent one-stage and two-stage detectors. Moreover, the FF-LPD-*3DConv* has the advantage of efficiency which is suitable for the real-time video LP detection.

The results on the SSIG-SegPlate dataset [69] are listed in Table II. The AP of proposed FF-LPD-*3DConv* and FF-LPD-*Trans* also exceed all compared video detectors. The FF-LPD-*3DConv* achieves competitive results (97.69 AP) compared to the best results of one-stage and two-stage detectors with high efficiency (48.4 FPS). The experimental results on these two datasets all show that our proposed FF-LPD method outperforms the compared video object detectors and reaches the fastest inference speed among all compared detectors.

In order to identify the performance intuitively, Fig. 8 displays the visualization results on some hard samples from UFPR-ALPR datasets. From Fig. 8, we find that DFF [56], FGFA [57], SELSA [75], and Temporal RoI Align [76] all have false detection and missing detection on these hard samples. And our proposed FF-LPD-*3DConv* and FF-LPD-*Trans* process these hard samples successfully. This prove that the proposed FF-LPD achieves better video detection accuracy and surpass other video detectors on efficiency. Meanwhile, the bounding boxes from FF-LPD-*3DConv* are more accurate compared to FF-LPD-*Trans* on these hard samples, which has the same conclusion in Table I (the average IoU of FF-LPD-*3DConv* is higher than that of FF-LPD-*Trans*).

In addition, the proposed FF-LPD-*3DConv* achieves the real-time LP detection under 48.4 FPS. The keyframe sample steps interval N is set to 5, which means the keyframe sample interval is only about 0.1 second in real-world scene. This small interval ensures the similarity between keyframe and non-keyframe, alleviating the pollution from memory features.

E. Ablation Studies

In order to judge whether the introduced modules have the promotion on LP detection task, we conduct ablation studies on UFPR-ALPR dataset to demonstrate the effectiveness of the proposed modules. The ablation studies results are illustrated in Table III.

First, we introduce the items in ablation studies (Table III). Frame-by-Frame means the frames are divided into keyframes and non-keyframes. Backbone-ResNet18 and Backbone-ResNet50 represent the backbone of the network. In case (c), the Frame-by-Frame, Backbone-ResNet18, and Backbone-ResNet50 are selected simultaneously, indicating that the detector uses ResNet-18 to obtain features from non-keyframes and uses ResNet-50 to extract features from keyframes. Knowledge Distillation is the proposed knowledge distillation strategy in Section III-B. Feature Propagation-*3DConv* is the designed 3D convolution based feature propagation method and the Feature Propagation-*Trans* is the transformer based method in Section III-C.

The baselines are case (a), case (b) and case (c) in Table III. From case (a), the ResNet-18 backbone only achieves 88.24 AP on UFPR-ALPR dataset. And the detector with ResNet-50 backbone, *e.g.* case (b), reaches 95.59 AP, which shows that the deeper backbone improves the detection performance dramatically. Meanwhile, the difference of efficiency between case (a) and case (b) is notable. Case (a) has 55.5 FPS, but case (b) only achieves 39.4 FPS, which means using a lightweight backbone can increase the inference speed. Case (c) introduces the frame-by-frame strategy, achieving 89.53 AP with 52.5 FPS. Because the majority of frames (non-keyframes) are processed by the lightweight low-level stream (ResNet-18), the frame-by-frame strategy maintains the inference efficiency effectively. Meanwhile, the increase of AP in case (c) is mainly because the high-level stream (ResNet-50) achieves higher accuracy on keyframes. Although case (c) arrives at fast inference (52.5 FPS), the AP of case (c) has obviously decrease compared to case (b). We introduce the knowledge distillation strategy into case (c), *i.e.* case (d), and the AP rises from 89.53 to 92.82, which verifies that the proposed feature-based knowledge distillation strategy improves the performance of low-level stream (ResNet-18) distinctly. The knowledge distillation strategy correctly instructs the low-level stream to mimic the high-level stream. In case (e), only the 3D convolution based feature propagation method is introduced into the baseline (case (c)). The AP of case (e) also increases from 89.53 to 89.84, which means the proposed 3D convolution based feature propagation method refines the low-level feature by spatial-temporal correlation and develops the detection performance to some degree. Meanwhile, we introduce the transformer based feature propagation method into

TABLE III
ABLATION STUDIES ON ACCURACY, RUNTIME AND COMPLEXITY BETWEEN OURS AND FLOW-WARPING METHODS.

Ablation Items	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
Frame-by-Frame			✓	✓	✓	✓	✓	✓
Backbone-ResNet18	✓		✓	✓	✓	✓	✓	✓
Backbone-ResNet50		✓	✓	✓	✓	✓	✓	✓
Knowledge Distillation				✓			✓	✓
Feature Propagation-*3DConv*					✓		✓	
Feature Propagation-*Trans*						✓		✓
AP	88.24	95.59	89.53	92.82	89.84	90.31	94.98	95.10
Average–IoU	0.7709	0.7891	0.7767	0.7415	0.7211	0.7356	0.7610	0.7340
FPS	55.5	39.4	52.5	50.5	48.6	15.9	48.4	15.9

Fig. 9. Some false predictions in UFPR-ALPR and SSIG-SegPlate datasets. The **red boxes** are ground truth, **green boxes** are false positive predictions, and **blue boxes** are correct predictions from our detector. The details are shown in the corner of the pictures. The filename is signed above the corresponding frame, and the symbol in brackets represents whether the frame is keyframe or not, according to the expression law in Section III-A.

case (c), *i.e.* case (f), and the AP increase from 89.53 to 90.31. This means the transformer based feature propagation builds better correlation between frames than 3D convolution based method. The transformer based method has huge potential for feature propagation. The proposed FF-LPD-*3DConv* approach (case (f)) reaches 94.98 AP, closing to the performance of case (b), and achieves a significant advance in efficiency, from 39.4 FPS in case (b) to 48.4 FPS in case (f). Eventually, the proposed FF-LPD-*Trans* (case (h)) achieve higher accuracy (95.10 AP) than FF-LPD-*3DConv*, but it is inefficient due to the consumption in transformer based feature propagation method, which means FF-LPD-*Trans* is unsuitable for real-time LP detection. The ablation studies demonstrate that FF-LPD-*3DConv* speeds up the inference and increases accuracy successfully.

In addition, we discuss the number of keyframe sample step length N in FF-LPD-*3DConv* which directly influences the trade-off between accuracy and efficiency in frame-by-

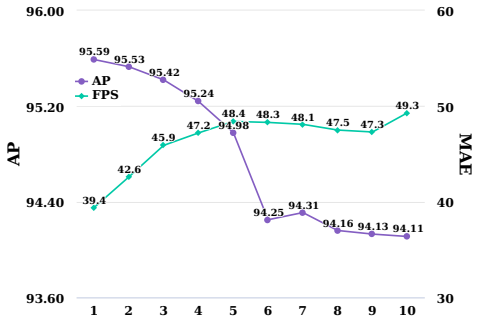


Fig. 10. The influence of keyframe sample interval N on accuracy and efficiency.

frame strategy. Fig. 10 shows the influence of N on the detection accuracy and inference speed. The detection accuracy decreases and the efficiency increases when the N increases. The small N will lead to more keyframes and these keyframes are processed by high-level stream which makes more accurate detection, but the efficiency decreases due to the time-consuming high-level stream. With the increase of N, there are more non-keyframes which are processed by low-level stream, and the low-level stream has faster inference speed than high-level stream, making the increase of efficiency. Moreover, when the $N = 6, ..., 9$, the efficiency even reduces compared to $N = 5$ because the Non-Maximum Suppression (NMS) causes more time than the case under $N = 5$. Until $N = 10$, the efficiency improves dramatically, but the accuracy also has significant reduction. Eventually, to balance the accuracy and efficiency, we choose $N = 5$ (94.98 AP and 48.4 FPS) in this paper.

F. Error Analysis

Fig. 9 demonstrates some false predictions in UFPR-ALPR and SSIG-SegPlate datasets. There are interesting findings in these errors. First, in these samples, our detector predicts the correct predictions (blue boxes), but it also considers some irrelevant contents as LP in some predictions (green boxes). From Track0098[22], the detector considers the traffic signs as LPs. In Track0106[10], Track0107[26], and Track0129[27], the detector judges the characters on the body of the car as LP. These mistakes show that the FF-LPD localizes the LP by judging whether characters are located on a uniform background. Meanwhile, the LP is usually fixed on the middle part of the bumper, and this clue helps the detector in localizing the LPs. However, a minority of vehicles do not follow this rule, such as trucks. In the UFPR-ALPR dataset, the LPs on trucks are set on one side of the bumpers instead of the middle, so that the detector believes that the slogans or symbols on the car body are LPs (in Track0106[10], Track0107[26], and Track0129[27]). From Track0131[2] and Track0144[3], we find that the detector also localizes the LP through colors. The LP in Track0131[2] is gray with the light color text, and the detector misjudges the gray car body as LP due to the similarity of colors. In Track0144[3], the detector considers the tail lamp as a red LP for bus. In addition, some false predictions in SSIG-SegPlate rely on the dataset itself. There are some unmarked LPs in the SSIG-SegPlate, and our detector also predicts those LPs without labels. Besides, in Track30[1] and Track31[5], the detector thinks of some vehicle appearances as LPs, which demonstrates that the detector also localizes the LP according to the texture with thick lines. This fact may be because the characters on the plate include abundant horizontal and vertical lines. In order to avoid these erroneous judgments, some prior knowledge of vehicles and LPs should be introduced to the LP detection task to alleviate the false positive samples. For example, the LP positions of truck and car are different, and a secondary processing model to recognize whether there are valid characters in predictions to filter the false predictions.

V. Conclusion

This paper presents the FF-LPD framework for video license plate detection, which improves the lightweight network performance in the video LP detection task and achieves real-time LP detection with accurate performance. Specifically, we introduce a feature-based knowledge distillation strategy and a feature propagation method based on spatial-temporal attention into the frame-by-frame video LP detector. The experimental results and ablation studies verify the effectiveness of components in the proposed FF-LPD. With the promotion of knowledge distillation and feature propagation method, the lightweight low-level stream for non-keyframe detection reaches the detection performance close to the high-level stream for keyframe detection, reducing the runtime and maintaining a high detection performance significantly.

In the proposed FF-LPD, the knowledge distillation strategy and feature propagation method can also be applied in other video detection tasks and added to other networks to achieve a high-performance real-time detection system. In addition, we will attempt to introduce the LP recognition task based on our LP detection method to achieve a real-time high-performance LP recognition system.

References

[1] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, and N. Komodakis, "A robust and efficient approach to license plate detection," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1102–1114, 2017.

[2] Q. Lu, W. Zhou, L. Fang, and H. Li, "Robust blur kernel estimation for license plate images from fast moving vehicles," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2311–2323, 2016.

[3] Q. Li, "A geometric framework for rectangular shape detection," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4139–4149, 2014.

[4] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, 2012.

[5] M. A. Rafique, W. Pedrycz, and M. Jeon, "Vehicle license plate detection using region-based convolutional neural networks," *Soft Comput.*, vol. 22, no. 19, pp. 6429–6440, 2018.

[6] M. Dong, D. He, C. Luo, D. Liu, and W. Zeng, "A cnn-based approach for automatic license plate recognition in the wild," in *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017.

[7] H. Li, P. Wang, and C. Shen, "Towards end-to-end car license plates detection and recognition with deep neural networks," *CoRR*, vol. abs/1709.08828, 2017.

[8] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA*. IEEE Computer Society, 2016, pp. 779–788.

[9] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA*. IEEE Computer Society, 2017, pp. 6517–6525.

[10] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[11] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020.

[12] G. Hsu, A. Ambikapathi, S. Chung, and C. Su, "Robust license plate detection in the wild," in *14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017, Lecce, Italy*. IEEE Computer Society, 2017, pp. 1–6.

[13] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*. IEEE, 2018, pp. 1–10.

[14] S. M. Silva and C. R. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *30th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2017, Niterói, Brazil, October 17-20, 2017*. IEEE Computer Society, 2017, pp. 55–62.

[15] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA*. IEEE Computer Society, 2014, pp. 580–587.

[16] R. B. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile*. IEEE Computer Society, 2015, pp. 1440–1448.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 8691, 2014, pp. 346–361.

[18] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, Quebec, Canada*, 2015, pp. 91–99.

[19] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy*. IEEE Computer Society, 2017, pp. 2980–2988.

[20] T. Vu, H. Jang, T. X. Pham, and C. D. Yoo, "Cascade RPN: delving into high-quality region proposal network with adaptive convolution," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada*, 2019, pp. 1430–1440.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 9905. Springer, 2016, pp. 21–37.

[22] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA*. IEEE Computer Society, 2017, pp. 936–944.

[23] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy*. IEEE Computer Society, 2017, pp. 2999–3007.

[24] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *CoRR*, vol. abs/1904.07850, 2019.

[25] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, Proceedings, Part XIV*, ser. Lecture Notes in Computer Science, vol. 11218. Springer, 2018, pp. 765–781.

[26] Z. Dong, G. Li, Y. Liao, F. Wang, P. Ren, and C. Qian, "Centripetalnet: Pursuing high-quality keypoint pairs for object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA*. IEEE, 2020, pp. 10 516–10 525.

[27] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: fully convolutional one-stage object detection," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South)*, 2019, pp. 9626–9635.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 3431–3440.

[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 12346. Springer, 2020, pp. 213–229.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[31] F. Liu, H. Wei, W. Zhao, G. Li, J. Peng, and Z. Li, "WB-DETR: transformer-based detector without backbone," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 2959–2967.

[32] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: unsupervised pre-training for object detection with transformers," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 1601–1610.

[33] T. Wang, L. Yuan, Y. Chen, J. Feng, and S. Yan, "Pnp-detr: Towards efficient visual analysis with transformers," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 4641–4650.

[34] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: deformable transformers for end-to-end object detection," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[35] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of DETR with spatially modulated co-attention," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 3601–3610.

[36] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional DETR for fast training convergence," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 3631–3640.

[37] W. Wang, J. Zhang, Y. Cao, Y. Shen, and D. Tao, "Towards data-efficient detection transformers," *CoRR*, vol. abs/2203.09507, 2022.

[38] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, ser. Lecture Notes in Computer Science, vol. 8693. Springer, 2014, pp. 740–755.

[39] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *CoRR*, vol. abs/1909.01754, 2019.

[40] G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Real-time automatic license plate recognition through deep multi-task networks," in *31st SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2018, Paraná, Brazil, October 29 - Nov. 1, 2018*. IEEE Computer Society, 2018, pp. 110–117.

[41] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and lstms," *CoRR*, vol. abs/1601.05610, 2016.

[42] C. Zhang, Q. Wang, and X. Li, "V-LPDR: towards a unified framework for license plate detection, tracking, and recognition in real-world traffic videos," *Neurocomputing*, vol. 449, pp. 189–206, 2021.

[43] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, Proceedings, Part XII*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11216. Springer, 2018, pp. 593–609.

[44] Silva, Sergio M. and Jung, Cláudio Rosito, "A flexible approach for automatic license plate recognition in unconstrained scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5693–5703, 2022.

[45] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[46] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA*. IEEE Computer Society, 2018, pp. 4320–4328.

[47] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 4793–4801.

[48] S. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA*. AAAI Press, 2020, pp. 5191–5198.

[49] X. Jin, B. Peng, Y. Wu, Y. Liu, J. Liu, D. Liang, J. Yan, and X. Hu, "Knowledge distillation via route constrained optimization," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South)*. IEEE, 2019, pp. 1345–1354.

[50] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Conference Track Proceedings*, 2015.

[51] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, Conference Track Proceedings*. OpenReview.net, 2017.

[52] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA*. IEEE Computer Society, 2017, pp. 7130–7138.

[53] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South)*. IEEE, 2019, pp. 1365–1374.

[54] S. Shin, J. Lee, J. Lee, Y. Yu, and K. Lee, "Teaching where to look: Attention similarity knowledge distillation for low resolution face recognition," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XII*, ser. Lecture Notes in Computer Science, vol. 13672. Springer, 2022, pp. 631–647.

[55] G. Chen, W. Choi, X. Yu, T. X. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA*, 2017, pp. 742–751.

[56] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA*. IEEE Computer Society, 2017, pp. 4141–4150.

[57] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy*. IEEE Computer Society, 2017, pp. 408–417.

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[59] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA*. IEEE Computer Society, 2018, pp. 7794–7803.

[60] F. Xiao and Y. J. Lee, "Video object detection with an aligned spatial-temporal memory," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, Proceedings, Part VIII*, ser. Lecture Notes in Computer Science, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11212. Springer, 2018, pp. 494–510.

[61] M. Shvets, W. Liu, and A. C. Berg, "Leveraging long-range temporal relationships between proposals for video object detection," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South)*. IEEE, 2019, pp. 9755–9763.

[62] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, Proceedings, Part XIII*, ser. Lecture Notes in Computer Science, vol. 11217. Springer, 2018, pp. 557–573.

[63] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA*. IEEE, 2020, pp. 10 334–10 343.

[64] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA*. IEEE Computer Society, 2018, pp. 3588–3597.

[65] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 8126–8135.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA*. IEEE Computer Society, 2016, pp. 770–778.

[67] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," *CoRR*, vol. abs/2111.06377, 2021.

[68] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA*. Computer Vision Foundation / IEEE, 2019, pp. 658–666.

[69] G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz, "Benchmark for license plate character segmentation," *J. Electronic Imaging*, vol. 25, no. 5, p. 053034, 2016.

[70] G. Jocher, "YOLOv5 by Ultralytics," May 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[71] C. Wang, A. Bochkovskiy, and H. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 7464–7475.

[72] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[73] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 13 039–13 048.

[74] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: exceeding YOLO series in 2021," *CoRR*, vol. abs/2107.08430, 2021.

[75] H. Wu, Y. Chen, N. Wang, and Z. Zhang, "Sequence level semantics aggregation for video object detection," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 9216–9224.

[76] T. Gong, K. Chen, X. Wang, Q. Chu, F. Zhu, D. Lin, N. Yu, and H. Feng, "Temporal ROI align for video object recognition," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 1442–1450.

[77] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Mmdetection: Open mmlab detection toolbox and benchmark," *CoRR*, vol. abs/1906.07155, 2019.

[78] MMTracking Contributors, "MMTracking: OpenMMLab video perception toolbox and benchmark," https://github.com/open-mmlab/mmtracking, 2020.

[79] MMYOLO Contributors, "MMYOLO: OpenMMLab YOLO series toolbox and benchmark," https://github.com/open-mmlab/mmyolo, 2022.

[80] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada*, 2019, pp. 8024–8035.

[81] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, ser. Lecture Notes in Computer Science, vol. 9912. Springer, 2016, pp. 483–499.

[82] S. Ji, Q. Ling, and F. Han, "An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information," *Comput. Electr. Eng.*, vol. 105, p. 108490, 2023.

Haoxuan Ding received the B.E. degree and the M.S. degree in aerospace propulsion theory and engineering from the Northwestern Polytechnical University, Xi'an, China, in 2018 and 2021 respectively. He is currently pursuing the Ph.D. degree from Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Junyu Gao received the B.E. degree and the Ph.D. degree in computer science and technology from the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China, in 2015 and 2021 respectively. He is currently a researcher with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Yuan Yuan (M'05-SM'09) is currently a Full Professor with the School of Computer Science and the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China. She has authored or co-authored over 150 papers, including about 100 in reputable journals, such as the IEEE TRANSACTIONS and PATTERN RECOGNITION, as well as the conference papers in CVPR, BMVC, ICIP, and ICASSP. Her current research interests include visual information processing and image/video content analysis.

Qi Wang (M'15-SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, pattern recognition and remote sensing.