Efficient One-stage Video Object Detection by Exploiting Temporal Consistency

Guanxiong Sun[1,2], Yang Hua[1], Guosheng Hu[2], and Neil Robertson[1]

[1] EEECS/ECIT, Queen's University Belfast, UK
[2] Oosto, Belfast, UK
{gsun02, y.hua, n.robertson}@qub.ac.uk, huguosheng100@gmail.com

Abstract. Recently, one-stage detectors have achieved competitive accuracy and faster speed compared with traditional two-stage detectors on image data. However, in the field of video object detection (VOD), most existing VOD methods are still based on two-stage detectors. Moreover, directly adapting existing VOD methods to one-stage detectors introduces unaffordable computational costs. In this paper, we first analyse the computational bottlenecks of using one-stage detectors for VOD. Based on the analysis, we present a simple yet efficient framework to address the computational bottlenecks and achieve efficient one-stage VOD by exploiting the temporal consistency in video frames. Specifically, our method consists of a location prior network to filter out background regions and a size prior network to skip unnecessary computations on low-level feature maps for specific frames. We test our method on various modern one-stage detectors and conduct extensive experiments on the ImageNet VID dataset. Excellent experimental results demonstrate the superior effectiveness, efficiency, and compatibility of our method. The code is available at https://github.com/guanxiongsun/EOVOD.

1 Introduction

Recently, in the field of object detection on still image data, great attention has been paid to one-stage detectors [7,17,26,31,38,39], as they have shown many stunning advantages compared to traditional two-stage detectors [3,9,11,27]. For example, one-stage detectors are more efficient, straightforward, and well aligned with other fully convolutional tasks, e.g., semantic segmentation, facilitating these tasks to share ideas and tricks. Given these advantages, many researchers work in different directions to further improve the accuracy of one-stage detectors, such as label assignment [6,37], feature alignment [4,22], loss design [18,20,35], and multilevel feature aggregation [8,19,21]. Until now, compared with two-stage detectors, one-stage detectors can achieve very competitive accuracy with faster run-time speed.

However, the field of video object detection (VOD) has been dominated by two-stage detectors [1,5,30,33,41,42] for many years and very little research has been investigated to transfer the merits of one-stage detectors from still images to videos [34]. This phenomenon contradicts empirical intuitions that one-stage detectors are more suitable for VOD task, which requires faster speed. To investigate the underlying reasons for this phenomenon, we conduct a comprehensive quantitative analysis (detailed in Section 3) and share the following facts: (1) The SOTA VOD methods apply attention-based feature aggregation to achieve promising speed-and-accuracy trade-offs; (2) In two-stage SOTA VOD methods, the computational cost is reasonable since the attention module take a small number of proposals as inputs, e.g., 300 proposals; (3) Directly adapting existing VOD methods to one-stage detectors introduces unacceptably high computations, due to the drastically increased number of inputs for attention modules, for example, 13k pixels in FCOS [31]; (4) For one-stage detectors on images or videos, the detection heads on low-level features take 80% computations.

On the basis of the aforementioned analysis, we propose two modules to achieve an efficient one-stage video object detector by fully taking advantage of the temporal consistency of video data. Here, the temporal consistency denotes the fact that objects change gradually in terms of *locations* and *sizes* in a sequence of consecutive frames. Inspired by this, we propose two novel modules, the location prior network (LPN) and the size prior network (SPN). Specifically, first, detected bounding boxes in the previous frame can guide the model to find regions where objects may appear in the current frame. Our LPN utilises this location prior knowledge to filter out background regions and thus reduces the computational cost. Second, objects keep in similar *sizes* within a short time. Another fact is that one-stage detectors divide objects into different levels of feature maps, and each level is responsible for detecting objects in a specific size range. Given the object sizes in the current frame, the proposed SPN enables our method to skip unnecessary computations on unrelated feature levels in several following frames.

In summary, our main contributions are:

- To our best knowledge, we are the first to investigate the obstacles to the development of one-stage VOD and conclude two bottlenecks causing high computations: very high-dimensional input for attention modules and unnecessary computations on low-level feature levels.
- We propose a simple yet effective framework to achieve efficient one-stage object detection. Specifically, a location prior network (LPN) filters out background regions and a size prior network (SPN) to skip computations on unnecessary feature levels. Note that our method can easily be incorporated into various one-stage detectors.
- Extensive experiments are conducted on ImageNet VID datasets with various one-stage detectors, i.e., FCOS [31], CenterNet [38] and YOLOX [7]. The results demonstrate that our method achieves superior speed-accuracy trade-offs and promising compatibility.

2 Related Work

One-stage Detectors. One-stage detectors can be classified into two categories. Firstly, key point based methods that predict pre-defined key points of objects to generate bounding boxes. For example, CornerNet [17] treats a bounding box as a pair of top-left corner and bottom-right corner and detects objects by grouping predicted corner pairs. ExtremeNet [39] predicts four extreme points (i.e., top-most, left-most, bottom-most, and right-most) and one center point to produce bounding boxes. CenterNet [38] detects the center point of an object bounding box. It predicts heat maps of center points and several regression values (i.e., center offset and size of the bounding box) to generate bounding boxes. In this paper, we use CenterNet as a representative of key point based one-stage detectors.

Another category of one-stage detectors is the center-based method, which regards the center pixels of an object as positives, and then predicts the distances from positives to bounding box boundaries. YOLO series [24,25,26] are the most well-known center-based one-stage detector. Recently, YOLOX [7] presents many empirical improvements to YOLO series, forming a new high-performance detector. DenseBox [14] utilises a filled circle located in the center of an object and predicts four distances from each location inside the circle to the boundaries of the object bounding box. FCOS [31] regards all locations inside an object as positives and introduces a centerness branch to measure distances between positives to the center point of the object. The centerness branch can effectively reduce false positives in the inference stage. In this paper, we use FCOS and YOLOX as representatives of center-based one-stage detectors.

Video Object Detection (VOD). VOD methods explore using temporal information within a video to improve the performance and the speed of single-frame detectors. Existing VOD methods can be divided into two categories: box-level methods and feature-level methods. Box-level methods try to refine the detection results using temporal associations of predicted bounding boxes. These methods are performed in a post-processing manner. For example, TPN [15] and TCNN [16] use LSTM and tracking to model temporal associations between detected bounding boxes. SeqNMS [10] extends NMS to the time domain and greatly reduces false positives. CHP [34] proposes a heat map propagation method for CenterNet [38], which makes detection results temporally smooth. In contrast, feature-level methods are investigated to improve the accuracy of video object detection by feature enhancement and can be trained end-to-end. FGFA [41], MANET [32] and THP [40] utilise optical flow to propagate and aggregate feature maps. SELSA [33], MEGA [1], LRTR [29] and RDN [5] enhance the instance features (proposals) of the current frame by reasoning the relationships between objects within a video via attention mechanisms.

3 Analysis of the Computational Bottlenecks in Attention-based One-stage VOD

The key reason why recent video object detection (VOD) methods [1,5,29,33] achieve state-of-the-art performance is the utilisation of attention mechanisms.

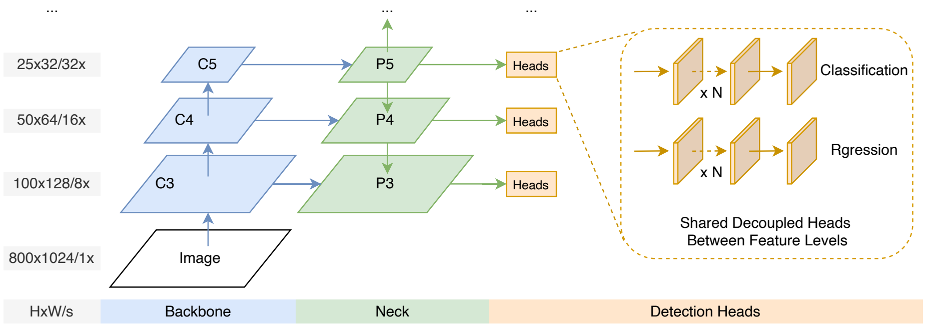


Fig. 1. General architecture of modern one-stage detectors, where H, W, and s are the height, width, and stride (down-sampling ratio) of feature maps, respectively. C3, C4 and C5 denote the output feature maps of the backbone. P3, P4, P5, etc. denote the feature levels in the neck, e.g., FPN. The decoupled detection heads, which usually contain a classification branch and a regression branch, are shared through all feature levels. *Best viewed in colour.*

However, these methods are based on two-stage detectors. We first apply attention-based VOD methods directly to one-stage detectors and demonstrate the high computational cost in this naive adaptation. Then, we conduct an elaborate analysis to locate the reasons for this high cost.

3.1 Preliminary Knowledge

Before we dive into a detailed analysis, we introduce some preliminary knowledge and define the necessary terms.

General Architecture of Modern One-stage Detectors. Modern one-stage detectors [7,20,26,31,38] are designed with different modules and settings, but they share a general architecture. The general architecture can be summarised as three parts: backbone, neck, and detection head. Specifically, backbone networks extract feature maps from input images, for example, ResNet-50/101 [12], DarkNet-53 [24], and HGNet [23]. Then, the feature maps $\{C\}$ are forwarded into the neck module, such as FPN [19] and PAN [21], to conduct multi-level feature aggregation. At last, the detection heads are performed on all feature levels $\{P\}$ to generate detections. A detailed general architecture is shown in Fig. 1, where s denotes the stride or down-sampling ratio of a feature level to the input image.

Complexity of the Attention Module. We introduce the complexity of the attention module because it is the key of designing an efficient one-stage VOD method. Given a query set $\mathbf{Q} = \{q_i\} \in \mathbb{R}^{N_q \times C}$ and a key set $\mathbf{K} = \{k_i\} \in \mathbb{R}^{N_k \times C}$, an attention module enhances each query q_i by measuring relation features as the weighted sum of all the keys in K. Here, N and C denote the number and the dimension of query or key elements, respectively. For simplicity, we use one-head attention for demonstration. Specifically, the enhanced feature of q_i is:

$$A(q_i, K) = q_i + \sum_j w_{ij} \cdot (W \cdot k_j), \tag{1}$$

where W denotes a linear transformation matrix, and w_{ij} is an element in the correlation matrix computed based on the similarity of all q-k pairs. Since the number of key elements N_k is usually equal or linearly related to the number of query elements N_q, and the complexity of an attention module is $O({N_q}^2 \times C)$, the computational cost of the attention module is very sensitive to N_q.

3.2 Naive Adaptation of Attention-based One-stage VOD

In SOTA VOD methods, attention modules are introduced in the second stage of two-stage detectors [3,27], where object proposals are treated as query elements. Specifically, the proposals of the current frame are considered as the query Q and proposals from reference frames are regarded as the key K for the attention module. Then, Q is enhanced with K by attention modules as Eq. (1). Since one-stage detectors generate proposals, a naive adaptation from SOTA methods to one-stage VOD is to conduct attention-based feature aggregation on the feature maps of one-stage detectors. Although the idea is straightforward, problems arise because of the difference between the number of proposals and the number of pixels on feature maps. For example, the number of proposals is quite small, e.g., 300 in FasterRCNN [27], while the number of pixels in one-stage detectors is usually thousands, e.g., ∼13K in FCOS [31]. This naive adaption highly increases the computational cost of the attention module.

We design several models to quantitatively demonstrate the increased computational cost problem of the naive adaptation method. Specifically, we use SELSA [33] as the baseline for its simplicity and promising performance. The analysis and conclusions in this subsection are suitable for other attention-based VOD methods because they have similar computation costs to SELSA. Following SELSA, the key set K is generated by randomly sampling n_r reference frames in the current video. The naive adaption is to treat all pixels on the feature maps as the query or key elements. For every time step, Q and K consist of pixels on feature maps of the current frame and the randomly sampled reference frames, respectively.

In the next two subsections, we analyse the experimental results and conclude two bottlenecks for the computational issue.

3.3 Bottleneck 1: Drastically Increased N_q

In Table 1, we show the computational cost of the SELSA baseline and naive adaptions with three one-stage detectors, denoted as FCOS^A, CenterNet^A, and

Table 1. Comparisons of N_q and computational costs between SELSA and directly applying attention mechanisms on one-stage detectors, FCOS, CenterNet, and YOLOX.

Method	Input Size	Strides $\{s\}$	N_q	GPU Mem.	FPS
SELSA	(600, 1000)	{16}	300	1.8 GB	18.5
FCOS^A	(600, 1000)	{8,16,32,64,128}	12,958	21.9 GB	4.6
CenterNet^A	(512, 512)	{4}	16,384	31.2 GB	4.3
YoloX^A	(640, 640)	{8, 16, 32}	8,400	16.7 GB	8.9

YOLOX^A. Although most SOTA VOD methods use more than 10 reference frames during inference, we only test with 2 reference frames because we suffer from GPU out-of-memory errors on Tesla V100 (32GB) GPU if more than 2 reference frames are used for naive one-stage adaption models. For example, in SELSA [33], $N_q = 300$ is the number of proposals of the current frame. Differently, in the naive one-stage adaption model, N_q is related to the size of the input image. For FCOS^A, we follow the protocols in SOTA VOD methods [1,5,33] to resize the input image to a shorter side being 600 and a longer side less or equal to 1000, and thus N_q is ∼13K. For CenterNet^A [38] and YOLOX^A [7], following the original papers, the input images are resized to 512 × 512 and 640 × 640, and thus N_q are ∼16.4K and 8.4K, respectively. Compared with SELSA, N_q of naive adaption models drastically increases nearly 50 times. As analysed in §3.1, the computational complexity of attention modules is quadratic to N_q. As a result, the GPU memory usage and the run-time speed of attention-based one-stage detectors are much larger and slower than SELSA, which makes them impossible to work in real-world applications.

To overcome this problem, one straightforward solution is to reduce the N_q for one-stage detectors. In two-stage detectors, RPN predicts proposals around the object regions to remove the background regions and thus produce a small number of proposals. However, RPN is removed in one-stage detectors, leading to a heavy computational cost. Here, we ask a question: can we utilise temporal information in videos to filter out background regions in a frame and reduce N_q? This question leads us to design the location prior network which uses the detection results of the previous frame to find possible foreground regions on the current frame.

3.4 Bottleneck 2: Detection Heads on Low Feature Levels

To further boost the speed and perform an efficient one-stage video object detector, we dissect the run time consumption in each part of the one-stage detector. We use FCOS as a representative for the demonstration. As shown in Table 2, the backbone and the neck module can run relatively fast. Nearly 80% of the running time is spent on the detection head. Then, we dissect the runtime consumption in the detection head according to feature levels. Specifically, around 65% of the running time is spent on the first feature level in the detection heads.

Table 2. Runtime dissection of a one-stage detector.

Part		Specification	Runtime (ms)	Ratio%
Backbone		R-50	7.7	18.0
Neck		FPN	0.5	1.2
Head	Level 1	s=8	27.7	64.9
	Level 2	s=16	3.9	9.1
	Level 3	s=32	1.4	3.3
	Level 4	s=64	0.8	1.9
	Level 5	s=128	0.7	1.6
All		-	42.7	100.0

The reason for this computational bottleneck is that the feature maps in low levels have very high resolutions. Therefore, the decoding process, i.e., generating detections for every location, is very time-consuming. The high-resolution feature maps are demonstrated to be useful for detecting small objects [19,21,31] and they inevitably consume huge computational costs.

However, it is possible to reduce the computational bottleneck on low-level feature maps by utilising the size prior knowledge of videos. Specifically, since the size of objects in consecutive frames changes gradually, we can skip the detection head on low-level feature maps for several frames if there is not any small object in the previous frame. Inspired by this observation, we design the size prior network and achieve a very fast one-stage VOD.

4 Methodology

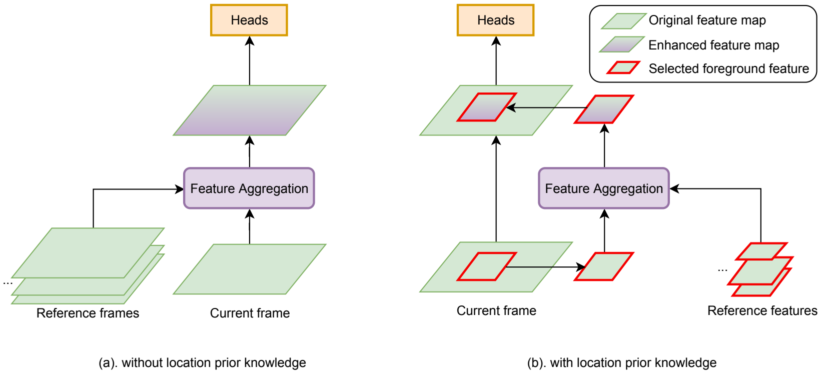
We introduce two modules, the location prior network (LPN) and the size prior network (SPN), to address the two computational bottlenecks, respectively. In this section, we illustrate the details in the LPN and SPN and FCOS [31] as a representative one-stage detector for demonstration.

4.1 Location Prior Network

As analysed in §3.3, reducing N_q is the key of performing an efficient one-stage video object detection with attention-based multi-frame aggregation. We propose a location prior network (LPN) to select foreground regions in the current frame to conduct partial feature aggregation. The LPN has two steps: First, the foreground region selection guided by the detected bounding boxes from the previous frame; Second, the partial feature aggregation to enhance the selected foreground pixels using attention modules.

We follow the open-source implementation[1] of SELSA[33] and use 14 random selected reference frames. The comparisons between conducting feature aggregation with and without LPN are shown in Fig. 2.

[1] https://github.com/open-mmlab/mmtracking



(a). without location prior knowledge

(b). with location prior knowledge

Fig. 2. (a) shows the process of directly conducting attention-based feature aggregation, where the purple rounded rectangle denotes the attention module. In (a), the input of the attention module is all pixels on the current frame and reference frames. (b) shows the pipeline of using location prior network for feature aggregation, where the red bounding box denotes the propagated bounding boxes from the previous frame. In (b), the input of the attention is foreground pixels on the current frame and the reference frames. *Best viewed in colour.*

Foreground Region Selection. Given detected bounding boxes of the previous frame, pixels within validated bounding boxes $\{D^v\}$ are regarded as foreground pixels for partial feature aggregation. Here, validated bounding boxes $\{D^v\}$ denote the detected boxes whose classification scores are greater than 0.5. If there is not a validated bounding box, the partial feature aggregation is skipped. Specifically, we project boxes in $\{D^v\}$ to each feature level by dividing the stride s of the level, e.g., 4 and 8. Then, we generate a binary mask $M \in \mathbb{R}^{1\times H\times W}$ of foreground regions for every level. The value of the mask at a location (x, y) is assigned as 1, if (x, y) falls into any validated bounding boxes. Otherwise, it is set to 0. In addition, before generating M, boxes in $\{D^v\}$ are resized with an adjustment ratio r to control the computational overhead.

Partial Feature Aggregation. Given the binary mask M, the pixel of location (x, y) is regarded as a foreground pixel if $M(x, y) = 1$. Then, foreground pixels are enhanced with features from reference frames $\mathbf{F}^r$ via attention modules. At last, the enhanced pixels are used to replace the pixels of the feature maps F in the same location. Specifically, the enhanced feature maps $\hat{F}$ are computed as follows:

$$\hat{F}(x,y) = \begin{cases} \mathbf{A}[F(x,y), \mathbf{F}^r] & \text{if } M(x,y) = 1, \\ F(x,y) & \text{else,} \end{cases} \tag{2}$$

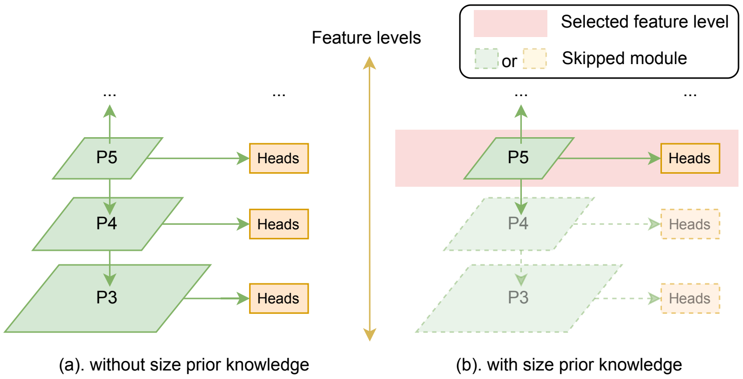


Fig. 3. (a) shows a normal detection process on multi-level feature maps where all levels of feature maps are passed to detection heads. (b) shows the detection process guided by the size prior network. The pink box denotes the feature level which the bounding boxes of the previous frame are generated from. In the current frame, computations on feature levels not in the pink box are skipped, denoted with the transparent boxes and dotted lines. *Best viewed in colour.*

where $\mathbf{A}(\cdot, \cdot)$ is defined as in Eq. (1) and (x, y) enumerates all locations on the feature maps. Finally, the new feature maps of the current frame are used for detection heads to predict bounding boxes.

Training and Inference. At the training stage, we adopt the strategy of temporal dropout used in [41] to randomly select two support frames within the same video of the current frame I_t. Then the ground truth boxes are used to generate the foreground mask and select query and key pixels. The whole network is optimised with the detection losses computed on the current frame in an end-to-end manner. During the inference stage, at a time step, the foreground mask of the current frame I_t is propagated from the detection results of the previous frame I_{t-1}. The key set consists of the pixels within detected bounding boxes on the reference frames.

4.2 Size Prior Network

The second computational bottleneck for efficient one-stage detection is due to the computations on low-level feature maps. We introduce the size prior network (SPN) to skip computations on low-level feature maps in unnecessary frames. Specifically, after the detection process of a frame at time step t, SPN selects validated bounding boxes $\{D_t^v\}$. Here, the validated boxes are obtained in the same way as mentioned in our LPN §4.1 (classification score > 0.5). For the next T time steps, the detection heads are conducted only on the feature levels that boxes in $\{D_t^v\}$ are generated from.

For example, in a video frame I_t, the validated bounding boxes $\{D_t^v\}$ are generated from the top feature level, e.g., P5, which indicates there might not be small objects in the following frames and thus the detection process is unnecessary to be conducted on low-level feature maps. For these frames, skipping the huge computations on low-level feature maps does not affect the detection accuracy. In practice, we set the detection frame interval T=7 for a good speed-accuracy trade-off. The comparisons between the detection process with and without SPN are shown in Fig. 3.

5 Experiments

In this section, we first conduct experiments to verify the effect of the proposed location prior network (LPN) and size prior network (SPN). We use three modern one-stage detectors as the base detector, i.e., FCOS [31], CenterNet [38], and YOLOX [7], which also demonstrate the good compatibility of our method. At last, we compare our method with SOTA video object detection methods.

5.1 Experimental Setting

Dataset. We evaluate our method on the ImageNet VID [28] dataset which contains 3,862 training and 555 validation videos. We follow previous approaches [1,33,41,42] and train our model on the overlapped 30 classes of ImageNet VID and DET set. Specifically, we sample 15 frames from each video in VID dataset and at most 2,000 images per class from DET dataset as our training set.

Backbone and Detection Architecture. For the backbone models, we use ResNet-101 [12] in FCOS and CenterNet, and DarkNet-53 [7] in YOLOX. For the neck architectures, FPN [19] and PAN [21] are used for FCOS and YOLOX, respectively. For CenterNet, following the same structure as in the original paper, the neck is built with 3 de-convolutional layers with 256, 128, and 64 channels, respectively. One 3×3 deformable convolutional layer is added before each deconvolution with channels 256, 128, and 64, respectively. In FCOS and YOLOX, shared detection heads are performed on all levels of feature maps to generate the detections. Specifically, there are two separate branches inside the detection head. Each branch contains four 3×3 convolution layers with 256 channels and one 3×3 convolution layer for predicting regression and classification result maps. In CenterNet, the detection head is built with a 3×3 convolutional layer with 64 channels followed by a 1×1 convolution with corresponding channels (e.g., number of classes) to generate detection outputs.

Training and Inference Details. We train all models on 4 Tesla V100 GPUs. For FCOS, images are resized to a shorter side of 600 pixels and the longer side less than or equal to 1000 pixels. With batch size equal to 4, we train the network for 3 epochs using the SGD optimizer (momentum: 0.9, weight decay: 0.0001).

Table 3. Effect of the LPN on different one-stage detectors.

Method	LPN	AP	AP50	AP75	APs	APm	APl	FPS
FCOS [31]		49.8	73.3.	54.6	10.1	22.4	56.1	25.1
FCOS [31]	✓	54.1	79.8	59.5	10.5	28.3	60.1	20.4
CenterNet [38]		49.3	73.4	55.4	10.2	16.3	56.5	40.9
CenterNet [38]	✓	53.4	79.8	58.5	10.3	21.9	59.5	35.5
YoloX-M [7]		49.4	69.4	55.4	11.1	25.0	55.4	39.7
YoloX-M [7]	✓	53.3	75.1	58.1	11.6	30.2	58.9	35.8

The learning rate is 10^{-3} for the first 2 epochs and 10^{-4} for the last epoch. For CenterNet, we follow [34] to resize the input images to 512 × 512. Random flip and random scaling from 0.6 to 1.4 are used as data augmentation and SGD is used as the optimizer. We train the network with a batch size of 32 and a learning rate of 10^{-4} for 50 epochs followed by a learning rate of 10^{-5} for 30 epochs. For YOLOX, we follow [7] to resize the input images to 640 × 640 and use additional data augmentations, including MixUp [36], Mosaic, RadomCrop, etc. We train the network with batch size 32 using the SGD optimizer. The initial learning rate is set to 10^{-3} with a cosine learning rate schedule for 80 epochs. In the inference phase, we resize the input image in the same way as in the training phase and reserve the top 100 confident detections per frame.

5.2 Effect of the Location Prior Network (LPN)

We first adapt LPN to three modern one-stage detectors to verify its effectiveness. Then, we conduct experiments using different bounding box adjustment ratios r in LPN to find an optimal speed-accuracy trade-off.

LPN on Various One-stage Detectors. To study how the proposed location prior network (LPN) influences the overall performance, we integrate LPN on three one-stage detectors and show the results in Table 3. LPN can improve the performance of all three detectors. Specifically, single-frame FCOS achieves 49.8% of AP. By utilising LPN to conduct multi-frame feature aggregation, the performance of FCOS+LPN is significantly improved by 4.3% to 54.1% of AP. Similarly, we can observe improvements in experiments of using LPN on more lightweight detectors, i.e., CenterNet and YOLOX-M. CenterNet+LPN and YOLOX-M+LPN boost their baseline performance from 49.3/49.4% of AP to 53.4/53.0% of AP, respectively. These experimental results demonstrate the effectiveness and compatibility of the LPN.

Effect of the Box Adjustment Ratios r. We show the experimental results on FCOS to study the effect of employing different bounding box adjustment ratios r. A smaller r results in fewer selected foreground pixels to be enhanced and thus leads to a faster speed. In contrast, a larger r selects more pixels to

Table 4. Effect of the bounding box adjustment ratios r in LPN.

r	AP	AP50	AP75	APs	APm	APl	FPS
0.5	53.6	78.5	57.9	9.6	27.0	58.5	21.4
0.8	54.1	79.8	59.5	10.5	28.3	60.1	20.4
1.0	54.2	80.0	59.7	10.5	28.5	60.2	19.1
1.2	54.2	80.0	59.8	10.6	28.4	60.2	17.5
1.5	54.2	79.9	59.9	10.9	28.4	60.2	14.7

Table 5. Effect of the SPN on different one-stage detectors.

Method	LPN	SPN	AP	AP50	AP75	APs	APm	APl	FPS
FCOS	✓		54.1	79.8	59.5	10.5	28.3	60.1	20.4
FCOS	✓	✓	53.8	76.9	58.9	9.8	27.3	59.5	26.9
YoloX-S	✓		53.3	75.1	58.1	11.6	30.2	58.9	35.8
YoloX-S	✓	✓	52.7	74.5	56.7	11.2	28.9	57.7	50.5

be enhanced but causes a slower speed. In our experiments, we vary r from 0.5 to 1.5. Our LPN achieves the optimal speed-accuracy trade-off when $r = 0.8$, i.e., 54.1% AP and 20.4 FPS. Once the adjustment ratio is larger than 0.8, the performance is less affected by the change in the adjustment ratios, but the run time keeps increasing. In our experiments, we set r as 0.8 by default.

5.3 Effect of the Size Prior Network (SPN)

Similar to the experimental design in §5.2, we conduct experiments by adding the size prior network (SPN) to two modern one-stage detectors to verify its effectiveness and compatibility. We use FCOS [31] and YOLOX [7] as representatives because they work with multi-level feature maps. Then, we conduct experiments using different frame intervals in SPN to find an optimal setting.

SPN on various One-stage Detectors. The location prior network (LPN) is designed to improve the accuracy of one-stage video object detection by conducting efficient multi-frame feature aggregation. While the size prior network (SPN) mainly focuses on improving the run-time speed by skipping the unnecessary computations in specific feature levels. Specifically, by introducing SPN, the run-time speed of FCOS+LPN and YOLOX+LPN are improved from 20.4/35.8 FPS to 26.9/50.5 FPS, respectively. At the same time, the accuracy is still at a comparable level with 53.8/52.7% of AP. As illustrated in §3.4, most computations of one-stage detectors happen on feature maps of low levels. Therefore, the run time saved by using SPN is mainly because of the computations saved in video frames where no small objects appear. To further understand the speed improvement, we list the portion of skipped frames and the portion of frames according to different object sizes in the supplementary material.

Table 6. Effect of temporal frame interval T in SPN.

T	AP	AP50	AP75	APs	APm	APl	FPS
0	54.1	79.8	59.5	10.5	28.3	60.1	20.4
7	53.8	76.9	58.9	9.8	27.3	59.5	26.9
14	53.0	75.4	56.6	9.6	26.9	58.7	28.4
21	51.9	73.8	55.7	9.2	25.4	56.9	29.0
28	48.5	73.3	54.0	8.9	24.6	55.4	29.5

Effect of the Temporal Frame Interval *T*. To explore the effect of temporal frame interval T in SPN, we show the performance of introducing SPN on FCOS+LPN under various T settings from 0 to 28 in Table 6. During inference, we first conduct a full detection on all feature levels of the current frame and then we use SPN to skip computations on some feature levels for T following frames. In the extreme case of $T = 0$, full detections are conducted on all frames. With $T = 7$, full detections happen in every 8 frames and partial detections happen in the 7 interval frames. In this setting, the run-time speed is significantly improved from 25.1 FPS to 40 FPS with the performance slightly decreased by 0.2% to 53.9% AP. By increasing the temporal interval T to larger numbers, the run-time speed continuously increases, however, the performance also degrades at the same time. In practice, we set the temporal interval $T = 7$ to obtain a good speed-accuracy trade-off.

5.4 Comparisons with SOTA Methods

We compare our method with SOTA VOD methods, and the results are shown in Table 7. As most SOTA methods are neither report with the run-time speed nor test on the same device, we re-implement recent SOTA methods and test them on our device for fair comparisons. The methods with $*$ denote our re-implementation versions. In addition, most SOTA methods are based on the two-stage detector, FasterRCNN, while we propose an efficient one-stage VOD method.

All results are reported with the same backbone R-101 except the YOLOX whose backbone is DarkNet-53. Overall, our method achieves a better speed-accuracy trade-off. In particular, our method with FCOS achieves 54.1% of AP at 20.4 FPS, which makes 0.7% accuracy improvement and nearly 3× speed improvement over the best competitor RDN. As expected, our method can achieve very good efficiency. Compared with the only existing one-stage VOD method, CHP, our method with CenterNet makes a 3.1% improvement of AP50. Considering the run-time speed, we adapt LPN and SPN on the YOLOX-M detector. Our method, YOLOX-M+LPN+SPN, runs very fast at 50.5 FPS on V100 GPU, much faster than other existing VOD methods. Moreover, the YOLOX-M+LPN+SPN model achieves good performance with 52.7% of AP, which is comparable to the SOTA method SELSA in our implementation. These results highlight the advantages of our method in terms of accuracy and speed. It is

Table 7. Comparisons with SOTA video object detection methods.

Method	Base Detector	AP	AP50	AP75	APs	APm	APl	FPS	Device
RDN [5]	FasterRCNN	-	81.8	-	-	-	-	10.6	V100
SELSA [33]	FasterRCNN	-	80.3	-	-	-	-	-	-
LRTR [29]	FasterRCNN	-	81.0	-	-	-	-	10	Titan Xp
MEGA [1]	FasterRCNN	-	82.9	-	-	-	-	8.7	2080ti
TFB [2]	FasterRCNN	-	83.8	-	-	-	-	4.9	2080ti
MAMBA [30]	FasterRCNN	-	**84.6**	-	-	-	-	9.1	Titan RTX
TransVOD [13]	Deform. DETR	-	81.9	-	-	-	-	-	-
CHP [34]	CenterNet	-	76.7	-	-	-	-	37	-
FasterRCNN* [27]	-	49.7	75.6	55.9	7.4	23.7	56.0	22.5	V100
FCOS* [31]	-	49.8	73.3	54.6	10.1	22.4	56.1	25.1	V100
CenterNet* [38]	-	49.3	73.4	55.4	10.2	16.3	56.5	40.9	V100
YoloX-M* [7]	-	49.4	69.4	55.4	11.1	25.0	55.4	39.7	V100
RDN* [5]	FasterRCNN	53.4	81.2	**60.1**	8.5	27.4	59.6	7.1	V100
SELSA* [33]	FasterRCNN	52.6	81.6	57.9	9.3	28.6	58.4	6.4	V100
MEGA* [1]	FasterRCNN	53.2	82.4	59.2	9.1	29.4	59.1	5.3	V100
Ours(+LPN)	FCOS	**54.1**	79.8	59.5	10.5	28.3	**60.1**	20.4	V100
Ours(+LPN)	CenterNet	53.4	79.8	58.5	10.3	21.9	59.5	35.5	V100
Ours(+LPN)	YOLOX-M	53.3	75.1	58.1	**11.6**	**30.2**	58.9	35.8	V100
Ours(+LPN+SPN)	YOLOX-M	52.7	74.5	56.7	11.2	28.9	57.7	**50.5**	V100

worth noting that, in our implementation, the two-stage FasterRCNN and the one-stage FCOS, CenterNet, and YOLOX achieve 49.7/49.8/49.3/49.4% of AP, respectively. The comparable performance of these base detectors demonstrates that the superior performance of our method is solely gained from the modules we proposed instead of the replacement of base detectors.

6 Conclusion

By comprehensive analysis, we indicate the computational cost is the underlying obstacle to achieving efficient one-stage video object detection. To address the computational bottlenecks, we propose a simple yet effective framework that can be incorporated into various one-stage detectors. Specifically, our method consists of two novel modules: (1) The location prior network that selects the foreground regions of the current frame using the detection results of the previous frame; (2) The size prior network to skip unnecessary computations on the low-level feature maps when there is not any small object appears. Extensive experiments are conducted, and excellent experimental results demonstrate the superior effectiveness, efficiency, and compatibility of our method.

References

1. Chen, Y., Cao, Y., Hu, H., Wang, L.: Memory enhanced global-local aggregation for video object detection. In: CVPR (2020)
2. Cui, Y., Yan, L., Cao, Z., Liu, D.: Tf-blender: Temporal feature blender for video object detection. In: ICCV (2021)
3. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: NeurIPS (2016)
4. Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L.: Dynamic head: Unifying object detection heads with attentions. In: CVPR (2021)
5. Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T.: Relation distillation networks for video object detection. In: ICCV (2019)
6. Ge, Z., Liu, S., Li, Z., Yoshie, O., Sun, J.: Ota: Optimal transport assignment for object detection. In: CVPR (2021)
7. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
8. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: CVPR (2019)
9. Girshick, R.: Fast r-cnn. In: ICCV (2015)
10. Han, W., Khorrami, P., Paine, T.L., Ramachandran, P., Babaeizadeh, M., Shi, H., Li, J., Yan, S., Huang, T.S.: Seq-nms for video object detection. arXiv preprint arXiv:1602.08465 (2016)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
13. He, L., Zhou, Q., Li, X., Niu, L., Cheng, G., Li, X., Liu, W., Tong, Y., Ma, L., Zhang, L.: End-to-end video object detection with spatial-temporal transformers. In: ACMMM (2021)
14. Huang, L., Yang, Y., Deng, Y., Yu, Y.: Densebox: Unifying landmark localization with end to end object detection. arXiv preprint arXiv:1509.04874 (2015)
15. Kang, K., Li, H., Xiao, T., Ouyang, W., Yan, J., Liu, X., Wang, X.: Object detection in videos with tubelet proposal networks. In: CVPR (2017)
16. Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., Zhang, C., Wang, Z., Wang, R., Wang, X., et al.: T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Transactions on Circuits and Systems for Video Technology **28**(10), 2896–2907 (2017)
17. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV (2018)
18. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In: NeurIPS (2020)
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
21. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018)
22. Lu, X., Li, Q., Li, B., Yan, J.: Mimicdet: Bridging the gap between one-stage and two-stage object detection. In: ECCV (2020)

23. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: ECCV (2016)
24. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
25. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: CVPR (2017)
26. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
28. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
29. Shvets, M., Liu, W., Berg, A.C.: Leveraging long-range temporal relationships between proposals for video object detection. In: ICCV (2019)
30. Sun, G., Hua, Y., Hu, G., Robertson, N.: Mamba: Multi-level aggregation via memory bank for video object detection. In: AAAI (2021)
31. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. arXiv preprint arXiv:1904.01355 (2019)
32. Wang, S., Zhou, Y., Yan, J., Deng, Z.: Fully motion-aware network for video object detection. In: ECCV (2018)
33. Wu, H., Chen, Y., Wang, N., Zhang, Z.: Sequence level semantics aggregation for video object detection. In: ICCV (2019)
34. Xu, Z., Hrustic, E., Vivet, D.: Centernet heatmap propagation for real-time video object detection. In: ECCV (2020)
35. Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N.: Varifocalnet: An iou-aware dense object detector. In: CVPR (2021)
36. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: Beyond empirical risk minimization. In: ICLR (2018)
37. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR (2020)
38. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
39. Zhou, X., Zhuo, J., Krähenbühl, P.: Bottom-up object detection by grouping extreme and center points. In: CVPR (2019)
40. Zhu, X., Dai, J., Zhu, X., Wei, Y., Yuan, L.: Towards high performance video object detection for mobiles. arXiv preprint arXiv:1804.05830 (2018)
41. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: ICCV (2017)
42. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: CVPR (2017)