Pattern Recognition 127 (2022) 108587

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Temporal-adaptive sparse feature aggregation for video object detection

Fei He[a,b], Qiaozhe Li[a,b], Xin Zhao[a,b,*], Kaiqi Huang[a,b,c]

[a] CRISE, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[c] CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, China

ARTICLE INFO

Article history:
Received 1 April 2021
Revised 22 December 2021
Accepted 11 February 2022
Available online 13 February 2022

Keywords:
Video object detection
Temporal-adaptive sparse sampling
Pixel-adaptive aggregation
Object-relational aggregation

ABSTRACT

Video object detection is a challenging task due to the appearance deterioration in video frames. To enhance feature representation of the deteriorated frames, previous methods usually aggregate features from fixed-density and fixed-length adjacent frames. However, due to the redundancy of videos and irregular object movements over time, temporal information may not be efficiently exploited using the traditional inflexible strategy. Alternatively, we present a temporal-adaptive sparse feature aggregation framework, an accurate and efficient method for video object detection. Instead of adopting a fixed-density and fixed-length window fusion strategy, a temporal-adaptive sparse sampling strategy is proposed using a stride predictor to encode informative frames more efficiently. A collaborative feature aggregation framework, which consists of a pixel-adaptive aggregation module and an object-relational aggregation module, is proposed for feature enhancement. The pixel-adaptive aggregation module enhances pixel-level features on the current frame using corresponding pixel-level features from other frames. Similarly, the object-relational aggregation module further enhances feature representation at proposal level. A graph is constructed to model the relations between different proposals so that the relation features and proposal features are adaptively fused for feature enhancement. Experiments demonstrate that our proposed framework significantly surpasses traditional dense aggregation methods, and comprehensive ablation studies verify the effectiveness of each proposed module in our framework.

1. Introduction

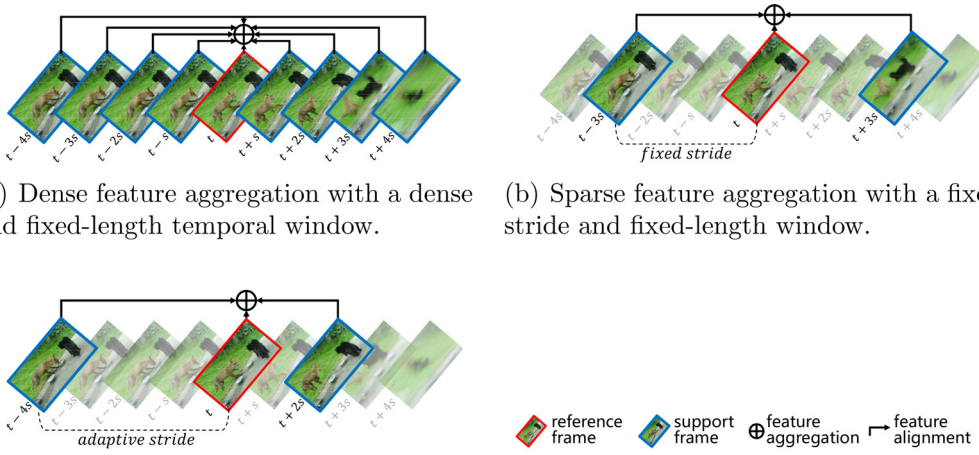
Video object detection is an important yet challenging computer vision task that has attracted increasing attention in recent years. In videos, object appearances are usually deteriorated by a variety of factors including motion blur, video defocus, or part occlusion, which are extremely challenging for image-based detectors. To address the appearance deterioration problem, video object detectors usually attempt to explore temporal information in videos to boost the detection performance on deteriorated frames.

One major solution of recent methods [1–3] is to aggregate features from nearby frames to exploit spatial-temporal coherence for feature enhancement of the deteriorated frames. Temporal aggregation is usually operated on a fixed-density and fixed-length temporal window. These related methods are hereinafter referred to as the dense feature aggregation (DFA) methods. However, such DFA strategies may be sub-optimal, and the reasons mainly lie in two aspects: (1) temporal information in videos could be extremely redundant so that dense sampling strategy might be computationally inefficient; (2) object appearances and locations may change irregularly over time and therefore additional noise might be also introduced by intuitive adjacent frame selection. In the following, we put these DFA methods in a unified view to explain how they operate (as shown in Fig. 1(a)).

As mentioned above, the DFA methods aggregate features of multiple adjacent frames to enhance the feature representation of the current frame. Specifically, for reference frame t in the video, the support frames to be aggregated are mechanistically sampled from a fixed-length and fixed-density temporal window [1–6]. Adjacent frames of video usually contain extremely redundant information, such as frames $t-4s$ to t in Fig. 1(a) are very similar. The feature representation of the current frame may only be slightly improved at the cost of massive inefficient computations. An intuitive way to reduce redundant computation is to sparsely sample the support frames with a fixed stride from the adjacent frames, as shown in Fig. 1(b). However, as objects may move irregularly over

* Corresponding authors.
E-mail addresses: hefei2018@ia.ac.cn (F. He), liqiaozhe2015@ia.ac.cn (Q. Li), xzhao@nlpr.ia.ac.cn (X. Zhao), kqhuang@nlpr.ia.ac.cn (K. Huang).

https://doi.org/10.1016/j.patcog.2022.108587
0031-3203/



(a) Dense feature aggregation with a dense and fixed-length temporal window.

(b) Sparse feature aggregation with a fixed stride and fixed-length window.

(c) Temporal-adaptive sparse feature aggregation method. Adaptively select informative adjacent frames to aggregate.

Fig. 1. (Best viewed in color) Comparison between our method and other multi-frame feature aggregation method. To obtain the detection results of reference frame t, the features of frame t would aggregate its nearby features for feature enhancement. Frame t is indicated as a reference frame, and the aggregated frames are indicated as support frames.

time, an object in one certain frame may be disturbed by objects from its adjacent frames. For example, in frame $t+3s$ of Fig. 1(b), the cat and fox become blurred after a sudden moving, which may introduce additional noise to frame t.

To exploit temporal information more efficiently, we propose a temporal-adaptive sparse feature aggregation framework for video object detection. Aiming at the drawbacks of the DFA methods, a stride predictor is proposed to adaptively sample informative nearby frames to be aggregated. As the appearances and locations of objects changing irregularly over time, the sampling strategy should adapt to the variation. As shown in Fig. 1(c), the cat and fox before frame t slowly move over time, and the adjacent frames contain a lot of redundant information. In this case, aggregating the distant frame $t-4s$ which contains more diverse information is more useful than other close frames. The cat and fox begin to move quickly after frame t, and the closer frame $t+2s$ should be more informative. Therefore, the stride predictor determines whether nearby frames should be sampled according to the motion speed of the current object. Compared with the DFA methods, our proposed method can achieve superior performance with fewer aggregated frames. Moreover, the proposed stride predictor can perform as a general module for video object detection frameworks and can be easily integrated into traditional DFA methods.

Furthermore, a collaborative feature aggregation framework, which consists of a pixel-adaptive aggregation module and an object-relational aggregation module, is proposed for feature enhancement. Pixel-adaptive aggregation module firstly enhances each pixel on the current feature map through aligned features from nearby frames. However, pixel-level feature alignment may be inaccurate when the appearance of objects severely deteriorates, which may adversely affect the subsequent feature enhancement. Therefore, an object-relational aggregation module is added to further enhance the features of the current frame. The proposal features extracted from both the current frame and the sampled frames are regarded as nodes of a specific graph. Each node of the graph aggregates information from its neighborhoods through graph convolution [7], and proposals are enhanced by the mined relation features for better detection. We argue that such high-level feature enhancement is a good complement to pixel-level feature enhancement, especially for severe appearance deterioration.

The contributions of this work are summarized below:

- A stride predictor is proposed to adaptively sample informative frames for feature aggregation. Compared with the traditional DFA methods, fewer aggregated frames can achieve higher detection accuracy.
- A pixel-adaptive aggregation module is proposed to achieve accurate pixel-level spatial alignment and efficient feature aggregation to enhance each pixel feature quality of the current frame.
- An object-relational aggregation module is further adopted to enhance the proposal features. A graph is proposed to model the object relations between different proposals for better feature enhancement.
- Our experiments show that the proposed model outperforms state-of-the-art methods on ImageNet VID [8], and the effectiveness of each proposed component is verified by comprehensive ablation studies.

This paper is based on and extends our conference version [9] in terms of several aspects. (1) Object-relational aggregation is added to exploit the proposal relations by graph reasoning to further enhance features of the current frame. (2) Considering that the object may move at different speeds before and after the current frame, the stride predictor estimates the bidirectional speeds instead of the single front speed to sample the front frames and the back frames separately. (3) In the conference version, the stride predictor and feature aggregation module are optimized separately in the training stage. we improve it to an end-to-end trainable framework, which makes the training process more convenient and fast. (4) More comprehensive experiments and analyses are presented.

2. Related work

2.1. Image-based object detection

Image-based object detection has achieved remarkable results on static images, with the significant progress of the deep Convolutional Neural Networks (CNNs) [10]. Image-based detectors are usually categorized into two genres, two-stage detectors, and one-stage detectors. The pipeline of two-stage detectors can be summarized as generating region proposals based on the extracted feature maps from the deep CNNs, and classifying and refining

the corresponding bounding boxes to obtain final detection results. Related works include R-CNN [11], Fast R-CNN [12], Faster R-CNN [13] and HON [14], etc. One-stage detectors directly predict the interested bounding boxes based on the feature maps. Related works include YOLO [15], SSD [16], RetinaNet [17], LAMD [18], etc. Recently, multi-scale feature learning [19,20], ensemble learning [21], and memory-based methods [22] in object detection have attracted a lot of attention. Wu et al. [23] proposes a multi-model fusion architecture to learn complementary deep features recursively to facilite salient object detection. Although the above mentioned methods work well on images, they cannot be directly utilized in video tasks because they cannot handle appearance deterioration problems.

2.2. Video object detection

Unlike static images, videos contain rich yet redundant temporal information, which makes video object detection and image-based detection quite different. Video object detection explores temporal information to boost detectors generally through two directions, object level, and pixel level.

To explore object level temporal information, Seq-NMS [24], T-CNN [25] and D&T [26] perform cross-frame bounding boxes linkages and then rescores the boxes associated with each linkage. ST-Lattice [27] detects on sparse key-frames and propagates the predicted bounding boxes to non-key frames through motion and scales. DorT [28] uses detector and tracker on key-frames and non-key frames respectively, to obtain detection results and track bounding boxes. STDnet-ST [29] adopts tubelet linking to link small objects across video frames for small object detection. All the above methods focus on bounding box association via independent processes of linking/tracking, which cannot be jointly optimized. RDN [3] and MEGA [30] augment the features of each object proposal by aggregating its relation features over the proposals from support frames in an end-to-end manner.

To explore pixel level temporal information, DFF [31] utilizes the optical flow network FlowNet [32] to estimate the per-pixel motion between two neighboring frames and propagate the features of the selected key-frames to neighboring non-key frames, reducing calculation and speeding up the whole framework. FGFA [1] also applies an optical flow network to align features, and the aligned features are used for feature aggregation to augment the features of reference frames to improve detection quality. THP [33] designs more advanced feature propagation and key-frame selection mechanisms to improve accuracy as well as speed. Different from previous works, STSN [2] applies a spatiotemporal sampling network instead of the optical flow network to perform frame-by-frame spatial alignment for aggregation. STMN [5] devises a MatchTrans module to achieve feature alignment and aggregates features with well-designed recurrent units. Chen et al. [34] proposes a long-term patchwise alignment method to estimate the long-term spatio-temporal constraint to facilitate salient object detection in short-term video contents. Chen et al. [35] proposes a 3DConv-based lightweight temporal unit, which can be inserted into each decoder layer to facilitate the interaction between spatial and temporal saliency cues. Chen et al. [36] proposes a universal learning scheme to further boost existing methods, which selects frames from the testing set according to semi-supervised motion quality perception to construct a new training set.

3. Methodology

3.1. Framework overview

An overview of the proposed framework is shown in Fig. 2. Each reference frame t aggregates support frames $t-b(t)$ and $t+a(t)$ to obtain detection results, where $a(t)$ and $b(t)$ are calculated by stride predictor. The features $\boldsymbol{f}_{t-b(t)}, \boldsymbol{f}_t$ and $\boldsymbol{f}_{t+a(t)}$ are obtained from a feature extractor (e.g., ResNet-101 [37]). Two feature aggregation modules are adopted to the sampled frames to enhance the feature representation of the reference frame. Pixel-adaptive feature aggregation module is first adopted to enhance each pixel feature on current frame. The DeformAlign module is proposed to handle spatial feature misalignment between $\boldsymbol{f}_{t-b(t)}, f_{t+a(t)}$ and $\boldsymbol{f}_t$, generating $\boldsymbol{f}_{t-b(t)\to t}, \boldsymbol{f}_{t+a(t)\to t}$, which are then aggregated by attention aggregation module to get $\boldsymbol{f}_{pixel}$. Object-relational aggregation module is then applied to the enhanced features $\boldsymbol{f}_{pixel}$. RPN [13] and RoIAlign [38] are utilized to generate object proposals $\boldsymbol{X}$ from $\boldsymbol{f}_{pixel}$ and support frames. A graph is constructed according to the similarity between different proposals. Each node on the graph aggregates information from its neighborhoods through graph convolution [7], and each proposal is enhanced by the mined relation features. Finally, the resulting enhanced proposals $\boldsymbol{Z}$ are then exploited for proposal classification and regression.

3.2. Temporal-adaptive stride predictor

To obtain diverse information at a reference frame t, previous methods [1–4,6] aggregate the long-term features of the input video frames based on a fixed-length sliding window. Extending the length of the sliding window can effectively increase the temporal receptive field size to obtain more temporal information. Nevertheless, considering the high redundancy of video, adjacent frames may comprise extremely redundant information, and the feature representation of the reference frame may only be slightly improved at the cost of massive inefficient computations.

Inspired by dilated convolution [39], we find that increasing the temporal stride between aggregated frames can increase the temporal receptive field without any computation increasing. The temporal stride s between two frames t_1 and t_2 in the same video is defined as $s = |t_2 - t_1|$. STMN [5] adopts a fixed temporal stride $s = 10$ at each reference frame to aggregate nearby frames. Since object appearances and locations may change irregularly over time, the fixed temporal stride strategy cannot model variable temporal information, and additional noise may be introduced by intuitive adjacent frame selection. A better temporal stride scheduling strategy should be adaptive to the varying dynamics in the temporal domain.

A natural criterion for judging the temporal stride at a reference frame is the motion speed of the object in the reference frame. Fast motion speed means that the target objects may move out of the screen after a short period of time, the framework should choose a smaller temporal stride and aggregate the closer frames for the reference frame. On the contrary, slow motion speed means the framework should choose a larger temporal stride and aggregate farther frames. The motion speed of an object is measured by its intersection-over-union (IoU) scores with its corresponding instances in the neighboring frames (e.g., ± 10 frames). The indicator is dubbed as 'motion IoU'. The lower the motion IoU is, the faster the object moves.

Therefore, a stride predictor is proposed adaptively selecting aggregated frames for each reference frame and the network details are shown in Fig. 2. $\boldsymbol{f}_t$ and $\boldsymbol{f}_{t-e}$ are features of two nearby frames from the same video, and e is a fixed value. The differences between $\boldsymbol{f}_t$ and $\boldsymbol{f}_{t-e}$, i.e., $\boldsymbol{f}_t - \boldsymbol{f}_{t-e}$, are taken as input of stride predictor, and the deviation score between frame t and $t-e$ is then predicted. The deviation score is formally defined as the motion IoU. Specifically, the prediction network comprises two convolutional layers with 3×3 kernel and 256 channels, a global average pooling, a fully-connected layer, and a sigmoid function that follows. We define the transformation between deviation score and temporal stride according to experiments. When the number of ag-

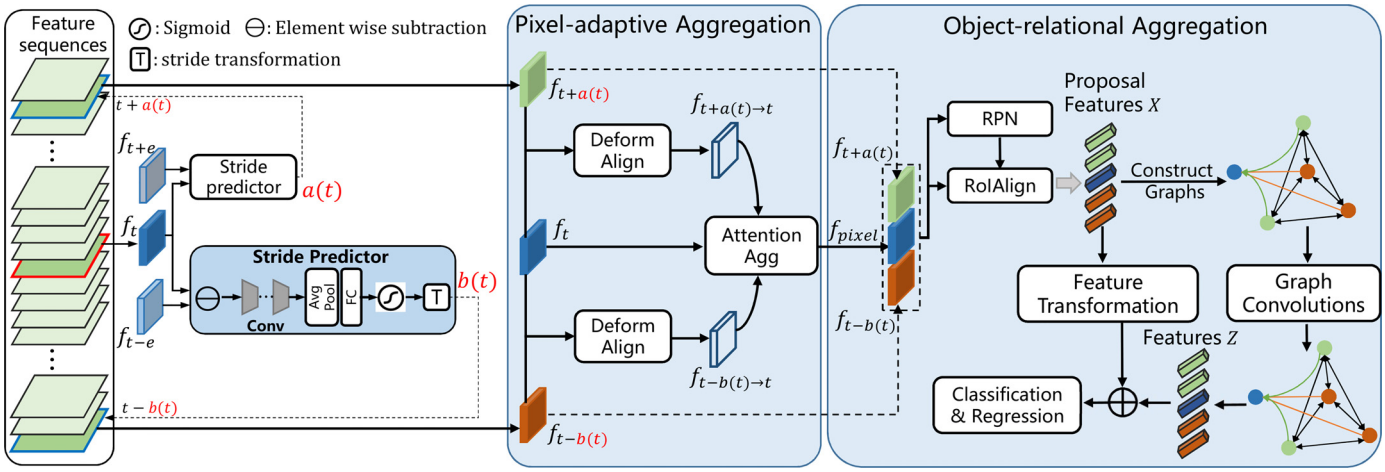


Fig. 2. (Best viewed in color) An overview of the proposed framework. First, given the input reference frames t and its nearby frames $t-e, t+e$, the stride predictor estimates the before and after motion speeds of the reference frame, and adaptively selects informative frames $t-b(t)$ and $t+a(t)$. Second, a pixel-adaptive aggregation module is adopted to enhance the reference features. The DeformAlign module aligns the features $\boldsymbol{f}_{t-b(t)}, \boldsymbol{f}_{t+a(t)}$ to $\boldsymbol{f}_t$. Then aligned features are aggregated by *Attention Agg* to enhance each pixel feature on the reference frame. Third, an object-relational aggregation module is adopted to the enhanced features f_{pixel}. RPN and RoIAlign are utilized to generate proposals $\boldsymbol{X}$ from $\boldsymbol{f}_{pixel}$ and support features. A graph is constructed according to the similarity between different proposals. Each node on the graph aggregates information from its neighborhoods through graph convolution. The features of each proposal are enhanced by the mined relation features. Finally, the enhanced proposals $\boldsymbol{Z}$ are exploited for proposal classification and regression.

gregated frames is 3 by default, if the predicted deviation is less than 0.7 (score < 0.7), reference frame t is set as fast temporal stride (10 by default). If the predicted score $\in [0.7, 0.9]$, frame t is set as middle temporal stride (24 by default). And the rest of the situation (score > 0.9), frame t is set as slow temporal stride (38 by default). When the number of aggregated frames increases, we adjust the temporal strides to keep the temporal receptive field unchanged. Different from our conference version [9], considering that the object may move at different speeds before and after the reference frame, $\boldsymbol{f}_{t-e}$ and $\boldsymbol{f}_{t+e}$ are sent to the predictor simultaneously with $\boldsymbol{f}_t$ to predict the front and back motion speeds of reference frame t, and then select the front frame $t-b(t)$ and back frame $t+a(t)$ for aggregation (as shown in Fig. 2).

3.3. Pixel-adaptive aggregation

3.3.1. DeformAlign feature alignment

Note that the appearance features of the same object are usually not spatially aligned across frames due to object motion. Any misalignment feature in the pixel-level feature aggregation may introduce artifacts around image structures, which may lead to false recognitions and inaccurate localization. Therefore, the DeformAlign module is proposed to utilize deformable convolution [40] to perform accurate pixel-level spatial alignment over time. The architecture of DeformAlign is shown in Fig. 3 **Left**.

In order to transform the support features $\boldsymbol{f}_i$ to align with reference features $\boldsymbol{f}_t$, DeformAlign first concatenates $\boldsymbol{f}_i$ and $\boldsymbol{f}_t$ as input to predict sampling parameters $\boldsymbol{\Theta}_i$ of feature $\boldsymbol{f}_i$, for each position p_k:

$$\boldsymbol{\Theta}_i(p_k) = f_\theta(\boldsymbol{f}_i, \boldsymbol{f}_t) = \{\Delta p_{k,n} | n = 1, \ldots, |R|\}, \tag{1}$$

where $R = \{(-1,-1), (-1,0), \ldots, (0,1), (1,1)\}$ donates a regular grid of a 3×3 kernel. With Θ_i and $\boldsymbol{f}_i$, the aligned features $\boldsymbol{f}_{i\to t}$ can be computed by the deformable convolution:

$$\boldsymbol{f}_{i\to t}(p_k) = \sum_{p_n \in R} \omega(p_n) \boldsymbol{f}_i(p_k + p_n + \Delta p_{k,n}). \tag{2}$$

The convolution operates on the irregular positions $p_n + \Delta p_{k,n}$, where $\Delta p_{k,n}$ may be fractional. To address the issue, the operation is implemented by bilinear interpolation, and details can be found in Dai et al. [40].

For the sampling parameter generation function f_θ, the concatenated $\boldsymbol{f}_i$ and $\boldsymbol{f}_t$ are reduced to 256 channels using two convolution layers with 3×3 kernel. After that, a 3×3 kernel with $2 \times k \times k$ channels is used to generate offsets, where k is the kernel size of the deformable convolution. In practice, an additional DeformAlign module is cascaded to enhance the transformation capability and further refine the coarsely aligned features.

3.3.2. Attention aggregation

Attention aggregation is proposed for feature aggregation after feature alignment, as shown in Fig. 3 **Right**. Different support frames are unequally informative and feature alignment may suffer from inevitable errors. Therefore, dynamical aggregating support frames at pixel-level are critical for effective feature aggregation. Inspired by the previous work [1], which indicates the importance of each support frame to the reference frame by adaptive weight, an attention module is used in aggregation to assign pixel-level aggregation weights on each frame.

Intuitively, at location p, if the aligned nearby features $\boldsymbol{f}_{i\to t}(p)$ are close to the reference features $\boldsymbol{f}_t(p)$, $\boldsymbol{f}_{i\to t}(p)$ should be paid more attention. The dot product similarity metric [41] is utilized to measure the similarity between the embedding features. The weights of the attention map are estimated by:

$$\boldsymbol{M}_t(p) = \sigma(\boldsymbol{f}^e_{i\to t}(p) \cdot \boldsymbol{f}^e_t(p)), \tag{3}$$

where σ is sigmoid function which restricts the output in [0,1], $\boldsymbol{f}^e = \varepsilon(\boldsymbol{f})$ and $\varepsilon(\cdot)$ is an embedding network to reduce the features to 256 channels using convolution layer with 3×3 kernel. The attention map $\boldsymbol{M}_t$ has the same spatial size with $\boldsymbol{f}_t$ and is then multiplied in a pixel-wise manner to the original aligned features $\boldsymbol{f}_{i\to t}$. These features which assign attention weights are concatenated and fused by a convolution layer with 1×1 kernel. The resulting features $\boldsymbol{f}_{pixel}$ contain information from the reference frame and support frames, and remain the same shape with $\boldsymbol{f}_t$.

3.4. Object-relational aggregation

Pixel-adaptive aggregation can improve the feature quality of each pixel. However, pixel-level feature alignment may be inaccurate when the appearance of objects is severely deteriorated, which

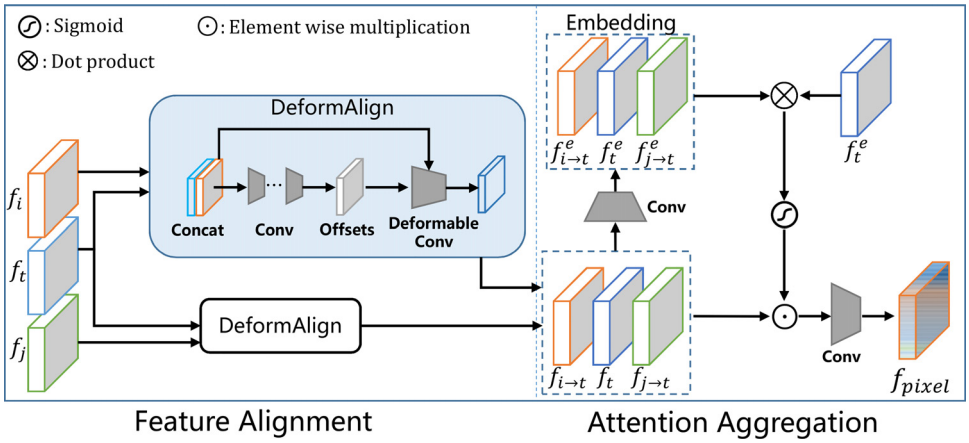


Fig. 3. Illustration of the proposed pixel-adaptive aggregation. **Left**: feature alignment module DeformAlign. **Right**: feature aggregation with attention weight.

may adversely affect the subsequent feature enhancement. Therefore, a graph-based module (as shown in Fig. 2) is designed to model the relations between different objects and further enhance the reference features at the object-level.

Given the enhanced features $\boldsymbol{f}_{pixel}$ and support features $\boldsymbol{f}_{t-b(t)}$, $\boldsymbol{f}_t$, $\boldsymbol{f}_{t+a(t)}$, RPN [13] and RoIAlign [38] are first utilized to extract the proposal features. The proposals from $\boldsymbol{f}_{pixel}$ are denoted as $\boldsymbol{X}^r = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_{N^r}] \in \mathbb{R}^{N^r \times D^{in}}$, and the proposals from support features are denoted as $\boldsymbol{X}^s = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_{N^s}] \in \mathbb{R}^{N^s \times D^{in}}$, where $\boldsymbol{x}_i$ is the ith proposal, N^r is the number of reference proposals, N^s is the number of support proposals, and D^{in} is the channel dimension of the proposal features. All the proposals are denoted as $\boldsymbol{X} = \boldsymbol{X}^r \cup \boldsymbol{X}^s \in \mathbb{R}^{N \times D^{in}}$, where $N = N^r + N^s$.

A graph is constructed to describe relation information between proposals. Each proposal in $\boldsymbol{X}$ is regarded as a node. $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of the graph, where $a_{ij} \in \boldsymbol{A}$ indicates the importance of jth node to ith node. In order to describe the relation between two nodes, the attention coefficients e_{ij} consider both the appearance features and position information, which are computed by $e_{ij} = g(f^a_{ij}, f^p_{ij})$, where f^a_{ij} denotes the appearance similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and f^p_{ij} denotes the position relation between node i and node j. We use embedding dot similarity [42] to measure appearance similarity:

$$f^a_{ij} = \frac{\alpha(\boldsymbol{x}_i)^T \beta(\boldsymbol{x}_j)}{\sqrt{d_k}}, \tag{4}$$

where α and β are two linear transformation functions that transform features into the embedding space $\mathbb{R}^{d_k}$. Positional encoding [42] is employed as position relation:

$$f^p_{ij} = \max\{0, \gamma(\varepsilon(\boldsymbol{x}^p_i, \boldsymbol{x}^p_j))\}, \tag{5}$$

where positions $\boldsymbol{x}^p_i$ and $\boldsymbol{x}^p_j$ are embedded to a high-dimensional representation by function ε, which computes cosine and sine functions of different wavelengths. The γ is a linear transformation function which transforms the embedded feature into a scalar weight. Inspired from graph attention layer [43], we perform masked attention and compute e_{ij} between node i and node j, $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the nodes in $\boldsymbol{X}^s$. Proposals in $\boldsymbol{X}^s$ are propagated on the graph to enhance reference proposals $\boldsymbol{X}^r$ and support proposals $\boldsymbol{X}^s$. The enhanced $\boldsymbol{X}^s$ are propagated on the next graph layer to further enhance each node. Attention coefficients e_{ij} are normalized by softmax function to compute element $a_{ij} \in \boldsymbol{A}$:

$$a_{ij} = softmax_j(e_{ij}) = \frac{f^p_{ij} exp(f^a_{ij})}{\sum_{j=1}^{\mathcal{N}_i} f^p_{ij} exp(f^a_{ij})}. \tag{6}$$

After constructing the graph, proposal features are propagated on the graph using the graph convolution [7] to mine the relations between different proposals. The outputs of the graph convolution are updated features of each node. Formally, one layer of graph convolution can be represented as:

$$\boldsymbol{Z} = \boldsymbol{AXW}, \tag{7}$$

where $\boldsymbol{A}$ is the adjacency matrix, $\boldsymbol{X}$ is the proposal features, $\boldsymbol{W} \in \mathbb{R}^{D^{in} \times D^{out}}$ is the trainable weight matrix, and $\boldsymbol{Z} \in \mathbb{R}^{N \times D^{out}}$ denotes the updated nodes features. The graph convolution can be stacked into multiple layers. Finally, the original features are transformed by a trainable linear transform function to fit the dimension of $\boldsymbol{Z}$, and outputs of the last graph layer are aggregated with the transformed features via summation to get the final outputs. The enhanced reference proposal features $\boldsymbol{Z}^r$ are exploited for proposal classification and regression to get final detection results.

3.5. Inference and training

3.5.1. Inference

Algorithm 1 is the detailed inference process of our model. Given an input video of consecutive frames $\{I_i\}$, the specified aggregation range K and the maximum temporal stride $s_{\max}$, minimum temporal stride $s_{\min}$. The proposed method sequentially processes each frame with a sliding feature buffer on the neighboring frames (of length $2Ks_{\max}+1$ in general, except for the beginning and the ending $Ks_{\max}$ frames). At initial, the feature network is applied in the beginning $Ks_{\max}+1$ frames to initialize the feature buffer and temporal stride (L3-L6 in Algorithm 1). Then the algorithm loops over all the video frames to perform video object detection, and to update the feature buffer. For frame I_i as the reference, the aggregated $2K$ frames are sampled at strides s_{bef} and s_{aft} from the feature buffer(L9-L16). Then the sampled features are aggregated by our pixel-adaptive aggregation module to get enhanced feature $\boldsymbol{f}_{pixel}$ (L17). RPN and RoIAlign are employed on $\boldsymbol{f}_{pixel}$ and original sampled features to generate proposal features (L18). A proposal graph is constructed based on these proposals and graph convolution is used to aggregate information from neighbor nodes (L19). The updated proposal features $\boldsymbol{Z}^r$ are fed to the detection network for object detection (L20). Finally, the feature maps are extracted on the $(i+Ks_{\max}+1)$th frame and are added to the feature buffer (L21–L22).

3.5.2. Training

The proposed framework is fully differentiable and can be trained end-to-end. Following the settings in the previous method [1], we randomly select T support frames $\{I_{t+s_1}, \ldots, I_{t+s_T}\}$

Algorithm 1 Inference algorithm of temporal-adaptive sparse feature aggregation for video object detection.

input: video frames $\{I_i\}$, aggregation range K, initialized temporal stride $s_{\min}$ and $s_{\max}$
$F = [\ ]$ ▷ feature buffer F
for $k = 1$ **to** $Ks_{\max} + 1$ **do** ▷ initialize F
  $\boldsymbol{f}_k = \mathcal{N}_{\text{feat}}(I_k)$
  $F.append(\boldsymbol{f}_k)$
  $s_k = s_{\min}$
end for
for $i = 1$ **to** ∞ **do** ▷ reference frame
  $A = [\ ]$ ▷ aggregated features buffer
  $A.append(\boldsymbol{f}_i)$
  $s_{i+10} = Stride(\boldsymbol{f}_i, \boldsymbol{f}_{i+10})$ ▷ predict stride
  $s_{bef}, s_{aft} = s_i, s_{i+10}$
  for $j = 1$ **to** K **do**
    $b(j), a(j) = \max(1, i - js_{bef}), i + js_{aft}$
    $A.append(\boldsymbol{f}_{a(j)}, \boldsymbol{f}_{b(j)})$
  end for
  $\boldsymbol{f}_{pixel} = PA\_Agg(A)$ ▷ pixel-adaptive aggregation
  $\boldsymbol{X}^r, \boldsymbol{X}^s = RoIAlign\&RPN(\boldsymbol{f}_{pixel}, A)$
  $[\boldsymbol{Z}^r, \boldsymbol{Z}^s] = OR\_Agg([\boldsymbol{X}^r, \boldsymbol{X}^s])$ ▷ object-relational aggregation
  $\boldsymbol{y}_i = \mathcal{N}_{\text{det}}(\boldsymbol{Z}^r)$ ▷ detect on the reference frame
  $\boldsymbol{f}_{i+Ks_{max+1}} = \mathcal{N}_{\text{feat}}(I_{i+Ks_{\max}+1})$
  $F.append(\boldsymbol{f}_{i+Ks_{max+1}})$ ▷ update F
end for
output: detection results $\{\boldsymbol{y}_i\}$

$(s_1, \ldots, s_T \in [-9, 9])$ from the adjacent frames of I_t. In our conference version [9], the stride predictor and feature aggregation module are optimized separately in the training stage, we improve it to an end-to-end trainable framework. During training, we use all $T + 1$ frames to train our model and take a pair of features $\boldsymbol{f}_t$ and $\boldsymbol{f}_{t+s_1}$ to optimize the stride predictor branch simultaneously. Here, the motion IoU between the input frame pair is computed as the regression target based on the ground truth objects. If there are multiple objects, calculate their average motion IoU. The training of the stride predictor can be carried out together with the detection network, and the training time is greatly reduced.

4. Experiments

4.1. Dataset and evaluation

We evaluate our model on the ImageNet VID [8], which consists of 3862 training videos and 555 validation videos from 30 object categories. We report mean Average Precision (mAP) on the validation set as the evaluation metric. Following the setting in Zhu et al. [1], both ImageNet VID and ImageNet DET [8] are utilized to train our model. Since the 30 object categories in ImageNet VID are a subset of 200 categories in ImageNet DET, the images from overlapped 30 categories in ImageNet DET are adopted for training.

4.2. Implementation details

We adopt ResNet-101 [37] as the backbone network and modify it slightly as in method FGFA [1]. Faster R-CNN [13] is utilized as our base detector. The pixel-adaptive aggregation is adopted on the top of *conv*5 stage. The fused features are split into 2 parts along axis 1. The first and the last part are fed to the RPN and detection head respectively. 128 proposals are sampled with a ratio of 1:3 for positive: negatives from the reference frame and 100 proposals with the highest objectness scores are sampled from each support frame. We implement the proposed framework mainly on Pytorch 1.7 [44]. We train our model in two stages. First, we pre-train our full model on the ImageNet DET dataset using the annotations of the 30 object classes that overlap with the ImageNet VID dataset. Note that ImageNet DET contains only images, and we use the reference frames as supporting frames. In the second training stage, the whole model is trained on ImageNet VID dataset, where the proposed networks are initialized from the weights learned in the first stage. For training, 120K iteration with SGD optimizer is performed on 4 NVIDIA RTX GPUs with each GPU holding one mini-batch. As mentioned before, each training batch contains $T + 1$ images. The learning rate begins with 10^{-4} and divides by 10 after 80K iteration. At inference, we adopt NMS with a threshold of 0.5 IoU to suppress reduplicate detection boxes.

Table 1
Performance comparisons with state-of-the-art video object detection models on ImageNet VID validation set.

Methods	Backbone	Base detector	#Frames	mAP (%)
FGFA [1]	ResNet-101	R-FCN	21	76.3
MANet [4]	ResNet-101	R-FCN	13	78.1
STSN [2]	ResNet-101+DCN	R-FCN	27	78.9
STMN [5]	ResNet-101	R-FCN	11	80.5
SELSA [45]	ResNet-101	Faster R-CNN	21	80.2
LLRTR [6]	ResNet-101	Faster R-CNN	33	81.0
RDN [3]	ResNet-101	Faster R-CNN	37	81.8
MEGA [30]	ResNet-101	Faster R-CNN	25	82.9
TCENet [9]	ResNet-101	R-FCN	3	80.3
Ours	ResNet-101	Faster R-CNN	3	82.5
Ours	ResNet-101	Faster R-CNN	7	83.4

4.3. Comparison with state-of-the-art methods

The performance of our method and other state-of-the-art video object detectors on the ImageNet VID validation set are shown in Table 1. Here we only list multi-frame aggregation methods that learn video object detectors by enhancing per-frame features from nearby video frames. For a fair comparison, we compare the backbone, base detector, and aggregated frames during evaluation. Almost all methods use ResNet-101 [37] as the backbone, except for STSN [2], which adds deformable convolution (DCN) [40] to the backbone and detector. R-FCN [46] and Faster R-CNN [13] are two commonly used base detectors with similar performance. TCENet [9] is our conference version model, which contains stride predictor and pixel-adaptively feature aggregation module. All the methods listed in Table 1 adopt a fixed-length window fusion strategy except SELSA [45] and our method. SELSA [45] randomly samples video frames for each reference frame to aggregate. Only our method adopts a smart temporal stride scheduling strategy that is adaptive to the varying dynamics in the temporal domain. Overall, our method has achieved better performance against other state-of-the-art methods under similar settings, and superior performance is offered with fewer aggregation video frames.

4.4. Experimental analysis

4.4.1. Ablation study

The effectiveness of each part in the proposed framework is also evaluated. To give a more comprehensive comparison, a variety of models are implemented. Table 2 compares our model with the image-based baseline and its variants.

Method (a) is the image-based baseline. It achieves 75.4% mAP with ResNet-101 and Faster R-CNN, which is close to the previous methods [3,30]. This indicates that our

Table 2
Accuracy and runtime of different methods on ImageNet VID validation.

Methods	(a)	(b)	(c)	(d)	(e)	(f)
pixel-adaptive		✓	✓	✓	✓	✓
object-relational			✓	✓	✓	✓
uniform sampling ($s = 20$)				✓		
stride predictor (ours)					✓	
stride predictor (aaai)						✓
mAP (%)	75.4	$79.0_{\uparrow 3.6}$	$80.4_{\uparrow 5.0}$	$82.0_{\uparrow 6.6}$	$82.5_{\uparrow 7.1}$	$82.3_{\uparrow 6.9}$
runtime (ms)	53	118	122	122	125	125

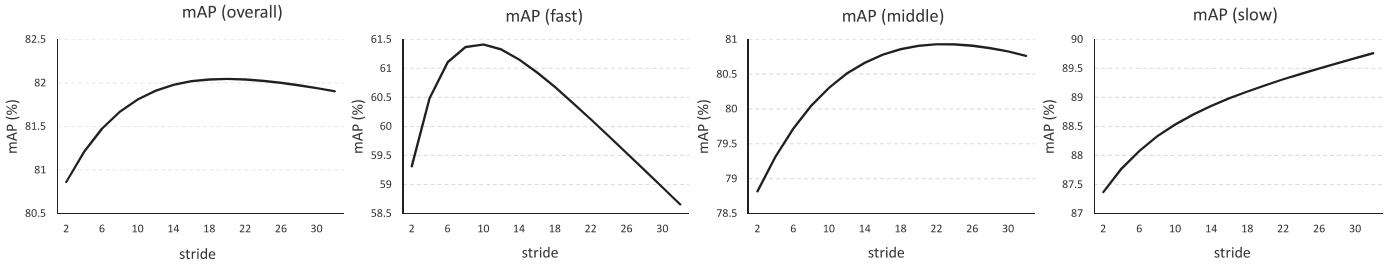


Fig. 4. Ablation study on temporal stride. The first subfigure is the results of the whole ImageNet VID validation, the last three subfigures are the results of the fast, middle, and slow motion subsets.

baseline is competitive and serves as a valid reference for evaluation. To verify the effectiveness of our method, we do not add bells and whistles like post-processing, model ensemble, etc.

Method (b) adopts pixel-adaptive aggregation in Fig. 3. The aligned features are aggregated with the attention module at pixel-level to enhance per-frame feature representation. After aggregating 3 frames (stride = 1), it increases the mAP score by 3.6% to 79.0%, which is comparable with other pixel-level feature aggregation methods [1,2,4]

Method (c) adds object-relational aggregation into method (b). Proposal features on the features $\boldsymbol{f}_{pixel}$ are augmented by proposals from support frames through a graph-based module. It further increases the mAP score by 1.4% to 80.4%, which demonstrates their complementarity.

Method (d) takes uniform sampling strategy for feature aggregation. After experiments with different fixed temporal strides, we find that the best performance can be achieved on the ImageNet VID validation set when the temporal stride is set to 20 (refer to the first line of Fig. 4). Compared with method (c), it does not bring extra time consuming, and the mAP score increased by 1.6% to 82.0%.

Method (e) adds the stride predictor to method (c). After adaptively selecting informative support frames to aggregate, it achieves 82.5% mAP, 2.1% higher than that of method (c). The results show that the adaptive sampling performed by stride predictor can achieve better performance when compared to taking uniform sampling strategy (method (d)). And in the case of image-based Faster R-CNN, there is a 7.1% mAP score increase, which indicates the effectiveness of our proposed framework.

Method (f) uses the conference version [9] stride predictor, which only predicts the front speed and samples the front and back aggregated frames with the same temporal stride. It is sub-optimal since the object moves at different speeds before and after the current frame. Therefore, we improve the stride predictor to predict the front speed and back speed of the reference frame simultaneously and sample the front frames and back frames separately. The experimental results show that this improvement is effective.

Table 3
Results of using stride predictor in other multi-frame aggregation methods.

Method	Stride predictor	#Frames	mAP (%)	runtime (ms)
FGFA [1]		21	76.3	256
FGFA [1]	✓	5	76.7	120
RDN [3]		37	81.7	190
RDN [3]	✓	5	81.9	135

4.4.2. Effect of temporal stride during inference

We conduct an experiment to study the influence of the temporal stride during inference. We achieve it by testing our model on different fixed temporal stride. For better analysis, besides the standard mAP scores on the whole ImageNet VID validation set, we also report the mAP scores over three subsets with different motion speeds. According to the motion IOU score, the objects are divided into fast (score < 0.7), middle (score $\in [0.7, 0.9]$), and slow (score > 0.9) subsets, respectively. The results are shown in Fig. 4. Increasing the temporal stride does not bring about any increase in computation. With the increase of temporal stride, the mAP score gradually boosts to a maximum point and then begins to decrease. It shows that increasing the temporal receptive field is quite effective for improving detection accuracy. However, the effective temporal receptive field length is limited. Too large temporal stride introduces invalid information and harms the result of feature aggregation. We can see that the faster the object motion speed is, the smaller the effective temporal receptive field is. Fast-moving objects should choose the close frame to aggregate, while slow-moving objects should choose the distant frame to aggregate. Therefore, a stride predictor can be effective for feature aggregation, and the transformation between the estimated motion IOU and temporal stride is set according to each maximum point in the three subfigures. We adopt the stride predictor in other multi-frame aggregation methods, and the results are shown in Table 3. After utilizing stride predictor in FGFA [1], we only aggregate 5 frames to achieve a similar performance (76.7% mAP) of the original aggregated 21 frames (76.3% mAP), and the processing time of

Table 4
Ablation study of using different motion threshold divisions in stride predictor.

Number	1	2	3	4
Motion threshold	[0, 1.0]	[0, 0.8], [0.8, 1.0]	[0, 0.7], [0.7, 0.9], [0.9, 1.0]	[0, 0.65], [0.65, 0.85], [0.85, 0.95], [0.95, 1.0]
Stride	20	14, 31	10, 24, 38	6, 21, 33, 42
mAP (%)	82.0	82.2	82.5	82.6

Table 5
Analysis on different validation sets.

Method	Faster R-CNN [13]	+ PA	+ OR	+ PA&OR
mAP^{all} (%)	75.4	80.9	81.2	82.5
mAP^{motion}_{slow} (%)	83.4	89.6	87.9	89.6
mAP^{motion}_{med} (%)	73.1	78.9	79.5	81.3
mAP^{motion}_{fast} (%)	51.7	59.4	60.3	61.1
mAP^{scale}_{large} (%)	84.6	90.1	89.8	91.5
mAP^{scale}_{med} (%)	48.6	59.2	62.9	64.7
mAP^{scale}_{small} (%)	22.5	30.8	33.1	35.6
$mAP^{occlusion}$ (%)	67.8	72.8	73.6	75.3

a single frame is greatly reduced (from 256ms to 120ms). Similar results are obtained in RDN [3]. We believe that the stride predictor is effective in other multi-frame aggregation methods.

4.4.3. Effect of different motion threshold divisions

We conduct an ablation experiment to explore the influence of different motion threshold divisions in stride predictor, and the results are shown in Table 4. We divide the motion IOU into different numbers of intervals, and set thresholds for each interval so that the number of ground truth objects contained in each interval is close. We use the method introduced in Section 4.4.2 to set the stride transformation for each interval. The smaller the motion threshold, the smaller the stride set. Denser division can set the temporal stride more flexibly and obtain better performance, but the process of exploring stride transformation is also more complicated. Establishing a continuous function mapping between motion IOU and stride may be a better stride transformation method, but it is very difficult in the irregularly changing video. Considering the complexity of stride transformation setting and performance, we set three interval divisions by default.

4.4.4. Effect of pixel-adaptive and object-relational aggregation

In order to demonstrate the effectiveness of pixel-adaptive and object-relational aggregation in our model, we conduct a more detailed analysis of the experimental results. The original validation set is divided into several subsets based on motion speed, object scale, and occlusion. The evaluation results are shown in Table 5. 'PA' means only pixel-adaptive aggregation with stride predictor is used. 'OR' means only object-relational aggregation with stride predictor is used. 'PA&OR' means the proposed temporal-adaptive sparse feature aggregation framework. The 3rd–5th rows in Table 5 show the results on subsets of different motion speeds. Compared with the image-based detector, both pixel-adaptive and object-relational aggregation methods have a high improvement in these subsets, especially in the fast motion speed subset. The 6th–8th rows in Table 5 show the results on subsets of different object scale. The objects are divided into small ($area < 50^2$ pixels), medium ($50^2 < area < 150^2$ pixels), and large ($area > 150^2$ pixels). Even if superior performance can be achieved on the whole dataset, small objects remain a big challenge. The object-relational aggregation provides a larger gain on small set. The last row in Table 5 shows the performance of occluded samples, which are sampled according to the occlusion annotations of ImageNet VID. Indeed, fast motion, small scale, and occlusion are extremely challenging for video object detection. Pixel-adaptive feature aggregation can improve the current results, but they are still affected by these severe appearance deteriorations. It's difficult to achieve accurate per-pixel correspondence during feature alignment. We argue that high-level features are more reliable to use when the object is small scale, fast-moving, or occluded. After performing both pixel-adaptive and object-relational aggregation, better results are achieved on almost all subsets, which shows that the two aggregation modules are complementary to each other and enhance the features collaboratively. Compared with the results of the image-based detector, our method has made significant improvement in all three subsets, which demonstrates the effectiveness of our method for appearance deterioration.

Table 6
Ablation study on the number of graph layers in object-relational aggregation.

Number	0	1	2	3	4
mAP (%)	75.0	75.7	76.0	75.8	75.4

Table 7
Performance comparisons by aggregating different number of frames.

#Frames	1	3	5	7	9
mAP (%)	75.4	82.5	82.8	83.4	83.7
runtime (ms)	53	125	191	285	357

4.4.5. Effect of the number of graph network layers

Table 6 compares the performance of using different number of graph network layers. In this experiment, ResNet-50 is utilized as backbone network, and the other settings are the same as method (c) of the ablation study. In the case of 0, the model only contains pixel-adaptive aggregation. With one graph layer, each proposal feature is enhanced with its first-order neighbors, and the mAP score is increased from 75.0% to 75.7%. When increasing the number of graph layers, the enhanced proposal features in the first layer are further used to enhance the proposal features in the next layer. The performance is getting better and tends to become flat. We speculate that two-order relation from neighbors is sufficient for most cases in ImageNet VID and high-order relation may introduce some unnecessary information. Hence, in our experiment, we set the number of graph layers as 2.

4.4.6. Effect of aggregated frame amount

The frames in aggregation are controlled by the aggregation range K in Algorithm 1. When the aggregation range is K, the number of aggregated frames is $2K+1$. We vary $2K+1$ from 1 to 9 to explore the effect of aggregated frame amount in our model, the results are shown in Table 7. We adjust the temporal stride to keep the temporal receptive field unchanged when the number of aggregated frames increases. The number of aggregated frames is 1 for the image-based detector. When the number of aggregated frames increases to 3 frames, the performance is greatly improved. Continue to increase the aggregated frames, the performance improves slightly, but the runtime is greatly increased. To achieve a balance between accuracy and runtime, we set the number of aggregated frames as 3.

4.5. Limitations

The proposed stride predictor introduces a novel idea to adaptively select the nearby frames to aggregate, and experiments show its effectiveness. However, the temporal stride setting in the stride predictor follows the assumption that the stride is positively correlated with the motion degree, which may be unstable in part of the scenes. Also, the transformation between the motion IOU and

temporal stride is hand-crafted, which may need to be reset when generalizing to other datasets. We hope our attempt can inspire the community to explore more robust and flexible adaptive selection strategies. Furthermore, since multi-frames are aggregated for each reference frame to boost the detection accuracy, the runtime of our method is slower than the image detector and hard to achieve real-time detection.

5. Conclusion and future work

We propose a temporal-adaptive sparse feature aggregation framework to effectively incorporate the temporal information for video object detection. Our main contributions are as follows, a stride predictor that adaptively selects support frames for the reference frame to aggregate, a collaborative feature aggregation framework, which consists of a pixel-adaptive aggregation module and an object-relational aggregation module, for feature enhancement. Experiments on ImageNet VID dataset have demonstrated the superiority of our proposed framework by comparison with other state-of-the-art video object detection methods. Ablation experiments show the effectiveness of each module. Thanks to the dynamic temporal information propagation strategy, the proposed method can significantly surpass traditional dense aggregation methods while aggregating fewer frames.

In future work, we plan to explore the potential of using our video object detection framework to develop other video understanding algorithms. Many video understanding tasks require the results of object detection, and our framework can be used as a strong object detection baseline. For example, in the task of weakly supervised video object segmentation [48–50], our framework can provide a more accurate object location result than image detectors, and the temporal information propagation contained in the object location process can be shared with the segmentation process for joint optimization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (Grant No. 61721004 and No. 61876181), the Projects of Chinese Academy of Science (Grant No. QYZDB-SSW-JSC006), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA27000000) and the Youth Innovation Promotion Association CAS.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patcog.2022.108587

References

[1] X. Zhu, Y. Wang, J. Dai, L. Yuan, Y. Wei, Flow-guided feature aggregation for video object detection, ICCV, 2017.
[2] G. Bertasius, L. Torresani, J. Shi, Object detection in video with spatiotemporal sampling networks, ECCV, 2018.
[3] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, T. Mei, Relation distillation networks for video object detection, ICCV, 2019.
[4] S. Wang, Y. Zhou, J. Yan, Z. Deng, Fully motion-aware network for video object detection, ECCV, 2018.
[5] F. Xiao, Y. Jae Lee, Video object detection with an aligned spatial-temporal memory, ECCV, 2018.
[6] M. Shvets, W. Liu, A.C. Berg, Leveraging long-range temporal relationships between proposals for video object detection, ICCV, 2019.
[7] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907(2016).
[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, CVPR, 2009.
[9] F. He, N. Gao, Q. Li, S. Du, X. Zhao, K. Huang, Temporal context enhanced feature aggregation for video object detection, AAAI, 2020.
[10] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[11] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR, 2014.
[12] R. Girshick, Fast R-CNN, ICCV, 2015.
[13] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, NeurIPS, 2015.
[14] J. Wang, X. Tao, M. Xu, Y. Duan, J. Lu, Hierarchical objectness network for region proposal generation and object detection, Pattern Recognit. 83 (2018) 260–272.
[15] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, CVPR, 2016.
[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, ECCV, 2016.
[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, ICCV, 2017.
[18] Q. Chen, P. Wang, A. Cheng, W. Wang, Y. Zhang, J. Cheng, Robust one-stage object detection with location-aware classifiers, Pattern Recognit. 105 (2020) 107334.
[19] W. Ma, Y. Wu, F. Cen, G. Wang, MDFN: multi-scale deep feature learning network for object detection, Pattern Recognit. 100 (2020) 107149.
[20] J. Yuan, H.-C. Xiong, Y. Xiao, W. Guan, M. Wang, R. Hong, Z.-Y. Li, Gated CNN: integrating multi-scale feature layers for object detection, Pattern Recognit. 105 (2020) 107131.
[21] J. Xu, W. Wang, H. Wang, J. Guo, Multi-model ensemble with rich spatial information for object detection, Pattern Recognit. 99 (2020) 107098.
[22] X. Li, M. Ye, Y. Liu, F. Zhang, D. Liu, S. Tang, Accurate object detection using memory-based models in surveillance scenes, Pattern Recognit. 67 (2017) 73–84.
[23] Z. Wu, S. Li, C. Chen, A. Hao, H. Qin, Recursive multi-model complementary deep fusion for robust salient object detection via parallel sub-networks, Pattern Recognit. 121 (2022) 108212.
[24] W. Han, P. Khorrami, T.L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, T.S. Huang, Seq-NMS for video object detection, arXiv preprint arXiv:1602.08465(2016).
[25] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, et al., T-CNN: tubelets with convolutional neural networks for object detection from videos, TCSVT, 2017.
[26] C. Feichtenhofer, A. Pinz, A. Zisserman, Detect to track and track to detect, ICCV, 2017.
[27] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. Change Loy, D. Lin, Optimizing video object detection via a scale-time lattice, CVPR, 2018.
[28] H. Luo, W. Xie, X. Wang, W. Zeng, Detect or track: towards cost-effective video object detection/tracking, AAAI, vol. 33, 2019.
[29] B. Bosquet, M. Mucientes, V.M. Brea, STDnet-ST: spatio-temporal ConvNet for small object detection, Pattern Recognit. 116 (2021) 107929.
[30] Y. Chen, Y. Cao, H. Hu, L. Wang, Memory enhanced global-local aggregation for video object detection, CVPR, 2020.
[31] X. Zhu, Y. Xiong, J. Dai, L. Yuan, Y. Wei, Deep feature flow for video recognition, CVPR, 2017.
[32] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, Flownet: learning optical flow with convolutional networks, ICCV, 2015.
[33] X. Zhu, J. Dai, L. Yuan, Y. Wei, Towards high performance video object detection, CVPR, 2018.
[34] C. Chen, G. Wang, C. Peng, X. Zhang, H. Qin, Improved robust video saliency detection based on long-term spatial-temporal information, TIP, 2019.
[35] C. Chen, G. Wang, C. Peng, Y. Fang, D. Zhang, H. Qin, Exploring rich and efficient spatial temporal interactions for real-time video salient object detection, TIP, 2021.
[36] C. Chen, J. Song, C. Peng, G. Wang, Y. Fang, A novel video salient object detection method via semisupervised motion quality perception, TCSVT, 2021.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[38] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, ICCV, 2017.
[39] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, ICLR, 2016.
[40] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, ICCV, 2017.
[41] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, CVPR, 2018.
[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, NeurIPS, 2017.
[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, ICLR, 2018.
[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch(2017).
[45] H. Wu, Y. Chen, N. Wang, Z. Zhang, Sequence level semantics aggregation for video object detection, ICCV, 2019.

[46] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, NeurIPS, 2016.
[48] L. Yang, J. Han, D. Zhang, N. Liu, D. Zhang, Segmentation in weakly labeled videos via a semantic ranking and optical warping network, TIP, 2018.
[49] D. Zhang, J. Han, L. Yang, D. Xu, SPFTN: a joint learning framework for localizing and segmenting objects in weakly labeled videos, TPAMI, 2018.
[50] J. Chen, Z. Li, J. Luo, C. Xu, Learning a weakly-supervised video actor-action segmentation model with a wise selection, CVPR, 2020.

Fei He received the B.Eng. degree in the department of automation from University of Science and Technology of China (USTC), Hefei, China, in 2018. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests include computer vision and deep learning.

Qiaozhe Li received the B.E. and M.S. degrees from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. He is currently a postdoctoral fellow at the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests include computer vision, deep learning, crowd understanding, and pedestrian analysis.

Xin Zhao received the Ph.D. degree from the University of Science and Technology of China. He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences (CASIA). His current research interests include pattern recognition, computer vision, and machine learning. He received the International Association of Pattern Recognition Best Student Paper Award at the ACPR 2011 and the 2nd Place Entry of the COCO Panoptic Challenge at the ECCV 2018.

Kaiqi Huang (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Nanjing University of Science Technology, China, and the Ph.D. degree from Southeast University. He is currently a Full Professor with Center for Research on Intelligent System and Engineering (CRISE), Institute of Automation, Chinese Academy of Sciences (CASIA). He is also with the University of Chinese Academy of Sciences (UCAS), and the CAS Center for Excellence in Brain Science and Intelligence Technology. He has published over 210 papers in the important international journals and conferences, such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Systems, Man, and Cybernetics, the IEEE Transactions on Circuits and Systems for Video Technology, Pattern Recognition, CVIU, ICCV, ECCV, CVPR, ICIP, and ICPR. His current researches focus on computer vision, pattern recognition, and game theory, including object recognition, video analysis, and visual surveillance. He serves as a Co-Chair and a program committee member over 40 international conferences, such as ICCV, CVPR, ECCV, and the IEEE workshops on visual surveillance. He is an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics: Systems and Pattern Recognition.