Class-aware Feature Aggregation Network for Video Object Detection

Liang Han, Pichao Wang, *Member, IEEE,* Zhaozheng Yin, *Senior Member, IEEE,* Fan Wang, *Member, IEEE,* and Hao Li, *Member, IEEE,*

***Abstract*—Recent progress in video object detection (VOD) has shown that aggregating features from other frames to capture long-range contextual information is very important to deal with the challenges in VOD, such as partial occlusion, motion blur, etc. To exploit more effective feature aggregation, we propose several improvements over previous works in this paper: (1) a class-aware pixel-level feature aggregation module, which characterizes a pixel by exploiting the context information lying in the instances from both the current frame and other frames. Different from the previous non-local operation, the proposed class-aware pixel-level feature aggregation filters out most of the noisy information from the large scope of background and objects in different classes, and only enhances representation of a foreground pixel with the same class instances with limited ambiguous information; (2) a class-aware instance-level feature aggregation module, which aggregates features for object proposals by learning two kinds of relations: the temporal dependencies among the same class object proposals from support frames sampled in a long time range or even the whole sequence, and spatial topology relation among proposals of different objects in the target frame. The homogeneity constraint in instance-level feature aggregation filters out many defective proposals, making the feature aggregation more accurate; and (3) a correlation-based feature alignment module embedded in the instance-level feature aggregation, which aligns the feature maps of the support and target proposals. Without bells or whistles, the proposed method achieves state-of-the-art performance on the ImageNet VID dataset without any post-processing methods. This project is publicly available** *https://github.com/LiangHann/Class-aware-Feature-Aggregation-Network-for-Video-Object-Detection.*

***Index Terms*—video object detection, class-aware, feature aggregation, pixel-level, instance-level, feature alignment**

I. Introduction

Due to the advancement of deep neural networks, significant progress has been achieved on object detection in still images [1], [2], [3], [4], [5], [6]. With the development of storage and communication, video is becoming a popular media to convey more rich information, and video-based analysis becomes inevitable. However, due to the deteriorated appearance caused by occlusion, motion blur, out-of-focus cameras, and rare poses in video capturing, directly applying those image-based object detectors on a frame-by-frame basis to a video often makes the performance unsatisfactory.

Recent research on VOD shows that it is useful to leverage the temporal information inherently encoded in videos to deal with the aforementioned challenges. Several works leverage short-term temporal information from nearby frames to help object detection in the current frame. For example, FGFA [7] and MANet [8] use optical flow to conduct feature aggregation, D&T [9] applies correlation features between nearby frames, STSN [10] adopts deformable convolutions across the temporal domain, and PSLA [11] explores the spatial correspondence between features across frames in a local region using progressive sparser strides. In those methods, only short-term temporal information is used, and the lack of long-term temporal information exploitation limits the detection performance of these methods, especially for objects with fast motion.

To take advantage of the long-term dependencies between frames, the recently proposed relation-based network [12] is widely adopted. Shvets et al. [13] propose to leverage Long-Range Temporal Relationship (LLR) to encode the inter-frame dependencies between object proposals in a long video segment, Wu et al. [14] introduce the Sequence Level Semantics Aggregation (SELSA) to further explore this long range relation in the sequence level, and Deng et al. [15] propose the Relation Distillation Networks (RDN) to progressively distill the long range relation. To leverage both the global and local temporal information, Chen et al. [16] design the Memory Enhanced Global-local Aggregation (MEGA) to better exploit the short- and long-term relations, and Jiang et al. [17] develop the Learnable Spatial-Temporal Sampling (LSTS) to mine the local motion information, and Sparsely Recursive Feature Updating (SRFU) and Dense Feature Aggregation (DFA) modules to exploit the global temporal information. To exploit the inter-video proposal relations, Han et al. [18] introduce the Hierarchical Video Relation Network (HVR-Net), by integrating intra-video and inter-video proposal relations in a hierarchical fashion.

There are several problems for most of the current relation-based feature aggregation methods of VOD. Firstly, most of them perform instance-level feature aggregation which inevitably overlooks the fine-grained pixel-level feature repre-

Manuscript received XX XX, 2020; revised XX XX, 2021. Corresponding author: Zhaozheng Yin. Equal contribution: Liang Han and Pichao Wang. Liang Han and Zhaozheng Yin have been supported by NSF grants CMMI-1646162 and CMMI-1954548.

L. Han is with Department of Computer Science, Stony Brook University, New York, USA (e-mail: liahan@cs.stonybrook.edu).

P. Wang is with Alibaba Group, Bellevue, USA (email: pichao.wang@alibaba-inc.com).

Z. Yin is with Department of Computer Science, Stony Brook University, New York, USA (e-mail: zyin@cs.stonybrook.edu).

F. Wang is with Alibaba Group, Sunnyvale, USA (email: fan.w@alibaba-inc.com).

H. Li is with Alibaba Group, Hangzhou, China (e-mail: lihao.lh@alibaba-inc.com).

sentation; secondly, they only consider the temporal dependencies among the objects, but neglect the spatial relations, which has been proved to be very useful in still image detection [12]; thirdly, all of these methods directly aggregate the support proposals in the temporal domain without considering whether they belong to the same class or not, making it inevitably bring defective proposals from irrelevant classes; lastly, these methods aggregate the features directly from support proposals without feature alignment, leading to unaligned features for the following regression and classification.

To exploit better feature aggregation for VOD, we propose a Class-aware Feature Aggregation network (CFA-Net) with the following improvements: (1) a Class-aware Pixel-level Feature Aggregation (CPFA) module, which enhances each pixel in the target feature map with all other pixels constrained in the same class instances of all support frames through self-adaptively predicted attention weights. Different from previous non-local operations [19], [20] which aggregate the global information, the proposed CPFA filters out a lot of noisy information from the large scope of background and different class instances, and only enhances pixel representation using object proposals with limited ambiguous information; (2) a Class-aware Instance-level Feature Aggregation (CIFA) network, which aggregates features for object proposals by learning two kinds of relations: the temporal dependencies among the same class objects from support frames sampled in a long time range or even the whole sequence, and spatial topology relation among proposals of different objects in the target frame. In CIFA, we separate spatial and temporal feature aggregation to distinguish the heterogeneity of temporal and spatial context information for instance-level feature aggregation. Moreover, the homogeneity constraint in CIFA helps filter out many defective proposals and only keep the object proposals which carry the same class label with the target proposal as the support proposals, which makes the feature aggregation more accurate; (3) a correlation-based feature alignment operation embedded in the instance-level feature aggregation, which aligns the support and target proposals that may have quite different poses, shapes, etc., making it more suitable for the following regression and classification step.

This paper is an extension of [21]. Compared with the ACM Multimedia 2020 conference paper, the extensions include: (1) pixel-level feature aggregation is proposed to overcome the drawback of the instance-level feature aggregation, i.e., enhance the fine-grained target pixel feature with the support pixel features; (2) a class constraint is added into the pixel-level feature aggregation module, which filters out most ambiguous information and keeps the most supportive and class-related pixel features to aggregate the target pixel feature; (3) the proposed class-aware pixel-level feature aggregation achieves superior performance on the widely-used VOD dataset; and (4) more experimental analyses are presented in this paper.

This paper is organized as follows. Section II reviews the related work on still image object detection and video object detection. Section III describes the proposed method. The experimental results are presented in Section IV, followed by the conclusion in Section V.

II. Related Work

A. Still Image Object Detection

There are two main branches for still image object detection: one-stage object detector and two-stage object detector. One-stage object detectors [2], [2], [22], [23], [4], [24] directly predict the bounding box of interest based on the feature map extracted by the backbone network. However, these methods usually lead to foreground and background class imbalance problem, which badly affects the training process [25]. Two stage detectors usually generate object proposals with a Region Proposal Network (RPN) [3] first, followed by a RoIAlign pooling [6] to get the proposal features, then the majority of negative proposals are filtered out, and the remaining proposal features are used to perform the detection with a classification layer and a regression layer. Two-stage detector is adopted in this paper.

B. Video Object Detection

There are two branches for video object detection. On the one hand, the redundancy in video frames can be leveraged to improve the detection speed. For example, Zhu et al. [26], [27] adopt optical flow to propagate the key frame feature to other frames to save the expensive feature extraction cost. Chen et al. [28] design a time-scale lattice to improve the speed with an extra classifier to re-score the bounding boxes. Liu et al. [29], [30] adopt Bottleneck-LSTM with MobileNet [31], [32] as the backbone and use SSD as the detector to improve the speed on the mobile devices. Similarly, Yao et al. [33] adopt object tracker for temporal propagation, and use reinforcement learning for adaptive key-frame scheduling. Xu et al. [34] propagate the previous reliable long-term detection in the form of heatmap to boost results of upcoming images for one-stage detector.

Temporal information encoded in videos can also be used to improve the performance of VOD, and our paper follows this trend. There are two major directions in exploiting temporal information. The first is focused on post processing [35], [36], [37]. These methods usually take the spatial and temporal coherence into consideration, and explore bounding box association rules across nearby frames to refine the per-frame detection results. Those methods are sub-optimal because they are highly dependent on the quality of initial detector which is trained without any temporal information. In contrast, the other category of methods [9], [7], [26], [28], [8], [38], [27], [10], [39], [11], [15], [13], [14], [40] exploits the temporal information in videos during training stage. Among these methods, optical flow based feature warping [41] is widely used to propagate the features across frames [7], [26], [8], [37]. However, the optical flow only exploits the temporal information between frames in short time range, and the warping does not work well in occlusion. To address these shortcomings, Guo et al. introduce PSLA [11] to model the spatial correspondence between features across frames in a local region using the progressive sparser stride, Tang et al. design a cuboid proposal network [40] that extracts spatio-temporal candidate cuboids and a short tubelet detection network that detects short tubelets in short video segments, Chen

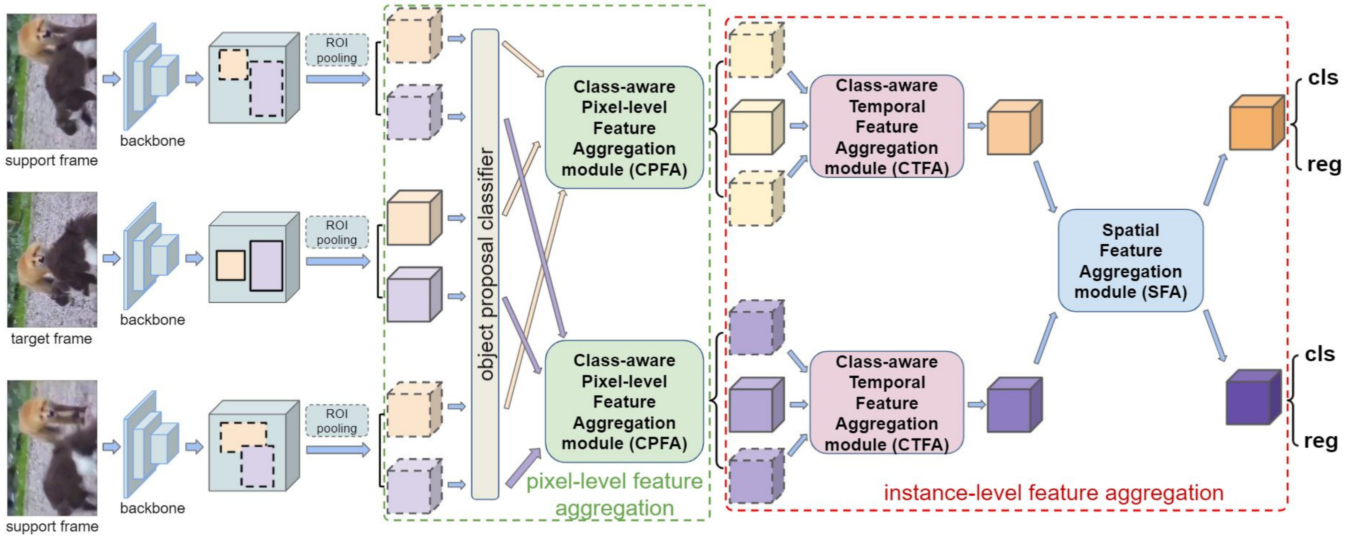
support frame
backbone
ROI pooling
target frame
backbone
ROI pooling
support frame
backbone
ROI pooling
object proposal classifier
Class-aware Pixel-level Feature Aggregation module (CPFA)
Class-aware Pixel-level Feature Aggregation module (CPFA)
pixel-level feature aggregation
Class-aware Temporal Feature Aggregation module (CTFA)
Class-aware Temporal Feature Aggregation module (CTFA)
Spatial Feature Aggregation module (SFA)
cls
reg
cls
reg
instance-level feature aggregation

Fig. 1: Flowchart of the proposed Class-aware Feature Aggregation Network (CFA-Net).

et al. develop a temporal refinement network (TRNet) and a temporal dual refinement network (TDRNet) [42] to propagate the refinement information across time. These methods only exploit the short-term temporal information, thus only limited support features are used to enhance the current feature. To explore the long-range dependencies in the temporal domain, Xiao and Lee [38] propose a spatial-temporal memory network (STMN) as the recurrent operation to model long-term temporal appearance and motion dynamics, with a MatchTrans module proposed to align the spatial-temporal memory. Shvets et al. [13] propose to use the relation module [43] to model the inter-frame dependencies between the object proposals in a long video segment, Wu et al. [14] further explore the temporal relation across the whole sequence, Deng et al. [15] propose the RDN to model the spatial-temporal relations for video object detection, and Chen et al. [16] design the MEGA to better exploit the short- and long-term temporal relations. To exploit the inter-video proposal relations, Han et al. [18] introduce the Hierarchical Video Relation Network (HVR-Net), by integrating intra-video and inter-video proposal relations in a hierarchical fashion. These works [13], [14], [15], [16], [17], [18] achieved promising results on video object detection. However, they are all instance-based feature aggregation scheme, and the performance largely depends on the quality of object proposals. Moreover, LLR [13] only aggregates the temporal instances, while SELSA [14], RDN [15], MEGA [16], and HVR-Net [18] treat all the instances equally and ignore the topology information of the proposals in the same frame.

III. Proposed method

To perform accurate object detection on deteriorated frames with partial occlusion, motion blur, or out-of-focus scene, a detector should be able to aggregate features of the same or the similar objects from other frames to enhance the appearance feature of the target object in the current frame. Moreover, the topology relation between different objects in the same frame can also help object detection and recognition. Keeping these motivations in mind, we propose a Class-aware Feature Aggregation network (CFA-Net) for VOD.

A. Overview

Fig. 1 presents an overview of the proposed CFA-Net. First, a backbone network (e.g., ResNet-101) is applied to extract features for the *target frame* (the current frame on which detection is performed) and the *support frames* (other frames in this video). Then, a RPN is adopted to generate object proposals for each frame, followed by a RoIAlign pooling operator to pool features for each object proposal. Before performing feature aggregation to enhance the target proposal features, we propose a coarse object proposal classifier to classify the generated object proposals, and the pixel- and temporal instance-level feature aggregation are only conducted on proposals with the same (predicted) class label. Specifically, the Class-aware Pixel-level Feature Aggregation module (CPFA) aggregates feature for each pixel of the pooled RoI feature map of each target proposal. Only the features of those pixels which are inside a proposal carrying the same class label as the target proposal can be used for feature aggregation, and such a constraint could filter out plenty of noisy and ambiguous support information. After pixel-level feature aggregation, for each target proposal, its feature is further enhanced by the features of other proposals at instance-level. Considering the heterogeneity of the spatial and temporal information, the instance-level feature aggregation is performed in these two dimensions separately. The Class-aware Temporal Feature Aggregation module (CTFA) is designed to enhance the target proposal features by aggregating proposal features with the same class label from support frames, in which the feature aggregation is guided by exploiting the appearance feature similarity between these proposals. The Spatial Feature Aggregation module (SFA) is designed to model the object topology relation by analyzing the interactions among objects in the same frame, and further aggregate features for the target proposal with the object proposals in the same target frame. Note that a Feature Alignment Module (FAM) is embedded in the temporal and spatial instance-level feature aggregation to align the feature maps of the target and support proposals. Finally, the aggregated target proposal features are input to two fully-connected layers to predict the class labels and regress the bounding box locations.

B. Object Proposal Classifier

Recently, the relation networks [12] and non-local neural networks [19] are proposed to explore the appearance and geometry relations among object proposals, and the feature aggregation is conducted based on the built relations between proposals. These two networks are adopted by many recent VOD works, and get promising performance. Unfortunately, the relations (i.e., the feature aggregation weights) are calculated mainly based on the proposal appearance feature, which inevitably includes some ambiguous information coming from the background or different kinds of objects when aggregating features for the target proposal, especially when the appearance feature of the target proposal is very similar to the ones of the background or other kinds of objects.

To perform feature aggregation with the most supportive and relative information, an object proposal classifier is designed and inserted in the detection network just before the feature aggregation modules. For the generated object proposals by RPN, the object proposal classifier will classify the proposals into different object classes or background. Then, the pixel-level feature aggregation and the temporal instance-level feature aggregation are performed among object proposals with the same (predicted) labels. This proposal classifier is jointly trained with the detection network. The benefits brought by the designed object proposal classifier are two-folds: first, with the predicted labels, the most supportive and relative features can be selected to perform feature aggregation; second, by training this proposal classifier, the features of object proposals from different classes extracted by the backbone network can be more distinguishable between each other.

C. Class-aware Pixel-level Feature Aggregation (CPFA)

Most of the recent works try to solve the challenging VOD task by instance-level aggregation only [13], [14], [15]. However, without accurate feature alignment, it is highly possible that a pixel in the target proposal is not aggregated with the most supportive ones. To better aggregate features for each target pixel, we propose to first perform feature aggregation in pixel level, i.e., characterizing each pixel by exploiting its contextual information in both the target frame and support frames to enhance its feature representation. Note that **pixel** here denotes a pixel location in the feature maps, which corresponds to a small region in the original image.

For a target pixel of an object, if we sample the support pixels without any constraint, the sampled support pixels may be from background or objects in different classes, and the aggregation blind to classes will degrade the representation of the target pixel. An example is presented in Fig. 2 to illustrate this problem. When performing feature aggregation for a pixel of a cat (red square in Fig. 2(a)), if the non-local neural network [19] is adopted to perform the pixel-level feature aggregation, the support pixels (blue squares in Fig. 2(b)) can be sampled from anywhere of a frame without any constraint. Thus, some pixels from background and different object classes might be leveraged to aggregate features for the target pixel, especially when the target object is very similar to the background or another object. This will increase the ambiguity of the representation of the target pixel, and harm the detection task. To perform feature aggregation for a target pixel with the most supportive ones, the selected support pixels should be restricted to belong to the same object or objects in the same class with the target pixel. Unfortunately, we can not get the exact object mask in this object detection task to put this instance constraint on the pixel-level feature aggregation. Instead, we use the proposal label predicted by the proposal classifier as a weak instance constraint, and restrict the selected support pixels (pink squares in Fig. 2(c)) to be in at least one of the proposals which have the same predicted labels with the target proposal. With this class constraint for the pixel-level feature aggregation, most background pixels and unrelated pixels coming from different object classes will be filtered out, which guarantees to sample the most supportive pixels for feature aggregation to the greatest extent.

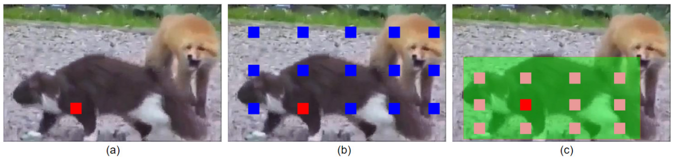

Fig. 2: Pixel-level feature aggregation without and with the instance constraint. For a target pixel (marked with red square), (b) shows the pixel-level feature aggregation without any constraint, i.e., support pixels (marked with blue squares) for feature aggregation can be sampled from anywhere of a frame, (c) presents the pixel-level feature aggregation with the class constraint, i.e., the sampled support pixels (marked with pink squares) for feature aggregation must be in at least one generated proposal which has the same predicted labels with the target proposal.

Let F denote the number of support frames, $H \times W \times C$ the size of feature tensors extracted by the backbone network with H, W and C as the height, width and channel dimension, respectively, the proposed pixel-level feature aggregation with class constraint can be formulated as

$$\mathbf{X}_{i,j} = \Phi^p\Big(\frac{1}{A}\sum_{k=1}^{H}\sum_{l=1}^{W}\sum_{f=1}^{F}\big(w^p(\phi^p(\mathbf{V}_{i,j}), \psi^p(\mathbf{V}^f_{k,l}))\cdot\mathbf{V}^f_{k,l}\cdot I^f_{k,l}\big)\Big) + \mathbf{V}_{i,j} \tag{1}$$

where $\mathbf{X}_{i,j}$ is the final aggregated feature value of the pixel at location (i,j) of the target frame; $\mathbf{V}_{i,j}$ is the original feature value of this target pixel before feature aggregation; $\mathbf{V}^f_{k,l}$ denotes the original feature value of the support pixel at location (k,l) of support frame f; Φ^p, ϕ^p and ψ^p are three different fully-connected layers; $A = \sum_{k=1}^{H}\sum_{l=1}^{W}\sum_{f=1}^{F} w^p(\phi^p(V_{i,j}), \psi^p(\mathbf{V}^f_{k,l}))\cdot I^f_{k,l}$, is a normalizing factor which serves as a softmax operation to preserve

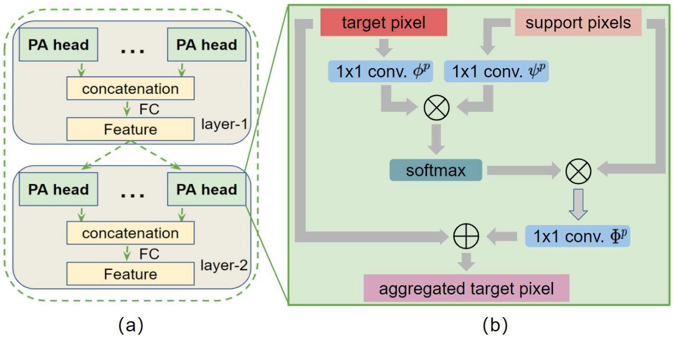

Fig. 3: Pixel-level feature aggregation. (a) The multi-head and multi-layer architecture adopted in the proposed CPFA, which includes 2 layers and 16 heads in each layer. (b) The detailed operations of the pixel-level feature aggregation in each PA head (Eq. 1). $\otimes$ denotes the matrix inner-product, and $\oplus$ is the element-wise addition.

the magnitude of the aggregated feature; $w^p(\cdot,\cdot)$ is the pixel relation (similarity) measure function defined as:

$$w^p(\phi^p(\mathbf{V}_{i,j}), \psi^p(\mathbf{V}^f_{k,l})) = \phi^p(\mathbf{V}_{i,j}) \odot \psi^p(\mathbf{V}^f_{k,l}) \tag{2}$$

where $\odot$ is the dot-product of two vectors. $I^f_{k,l}$ is an indicator function defined as:

$$I^f_{k,l} = \begin{cases} 1 & \text{if } \mathbf{V}^f_{k,l} \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where Ω denotes the set of all pixels of support proposals carrying the same predicted class with the target proposal.

Similar to [44], we design a multi-head and multi-layer architecture for the CPFA, which is depicted in Fig. 3 (a). There are two layers in the CPFA and each layer consists of D (D is 16 in our experiments) Pixel Aggregation heads (PA head). Fig. 3 (b) illustrates the detailed operations of the pixel-level feature aggregation in each PA head. To keep the CPFA in-place (i.e., input and output with the same feature dimension), at the very beginning of each layer, we first use a 1×1 convolutional layer $conv$ to reduce the input feature dimension to $\frac{1}{D}$ of the original dimension before feeding it into each PA head. The outputs of each PA head are then concatenated together to get back the original dimension, and the concatenated feature is then fed into the next layer. This operation is performed in each layer of the CPFA.

D. Instance-level Feature Aggregation

The class-aware pixel-level feature aggregation is performed on each pixel of the extracted proposal feature map, which guarantees us to aggregate the representation of a pixel in the target proposal with the most supportive pixels in the support proposals which are in the same class with the target proposal. However, the pixel-level feature aggregation can not solve the occlusion problem effectively, i.e., if an object is partially occluded by some other objects in a frame, the original object information in the occluded part can not be compensated by pixel-level feature aggregation because of the lack of the holistic representation of object. Therefore, after performing the pixel-level feature aggregation, we propose to further aggregate proposals features on the instance level. The instance-level feature aggregation consists of two steps: the Class-aware Temporal Feature Aggregation (CTFA) and the Spatial Feature Aggregation (SFA), the CTFA is performed on the pixel-level aggregated proposal feature, i.e., the output of the class-aware pixel-level feature aggregation, and the SFA is performed on the output of the CTFA. The detailed process of the instance-level feature aggregation can be found in Fig. 1 (the instance-level feature aggregation part).

Leveraging the holistic abstraction of an object to perform instance-level feature aggregation is a feasible way to deal with occlusions in VOD. Considering the heterogeneity of the temporal and spatial feature for aggregation [45], [21], the instance-level feature aggregation is separately performed in temporal dimension and spatial dimension. Different from [45] which performs feature aggregation first on spatial dimension then on temporal dimension to generate the tracklet representation with the help of instance ID, which is their final goal, our method adopts an opposite order, as the proposals from the temporal domain are more reliable and can enhance the feature for spatial relation exploration. This will be verified by some experiments (in Sec. IV-D).

1) Class-aware Temporal Feature Aggregation (CTFA): To select the most informative and relative support proposals for the temporal instance-level feature aggregation, the class constraint we used in the pixel-level feature aggregation is also added here. More precisely, for a target proposal, only the support proposals that are with the same (predicted) class label are picked to perform the feature aggregation.

The CTFA designed for temporal instance-level feature aggregation adopts the same architecture with the CPFA for pixel-level feature aggregation (i.e., multi-head and multi-layer architecture, which is shown in Fig. 3 (a)). For a target proposal p^t, let $P^s = \{p^s_1, p^s_2, ..., p^s_M\}$ be the support proposal set, $\mathbf{X}^t$ and $\mathbf{X}^s_m$ $(m = 1, 2, ..., M)$ be the feature maps of the target proposal and the support proposals, respectively. Mathematically, the temporal instance-level feature aggregation performed in a Temporal Aggregation head (TA head) is represented as:

$$\begin{aligned}\mathbf{Y}^t &= \Phi^t\Big(\sum_{m=1}^{M} softmax(w^a(\mathbf{X}^t, \mathbf{X}^s_m)) \cdot \mathcal{A}(\mathbf{X}^s_m)\Big) + \mathbf{X}^t \\ &= \Phi^t\Big(\sum_{m=1}^{M} \frac{exp\big(\phi^t(\mathbf{X}^t) \odot \psi^t(\mathbf{X}^s_m)\big)}{\sum_{j=1}^{M} exp\big(\phi^t(\mathbf{X}^t) \odot \psi^t(\mathbf{X}^s_j)\big)} \cdot \mathcal{A}(\mathbf{X}^s_m)\Big) + \mathbf{X}^t,\end{aligned} \tag{4}$$

where Φ^t, ϕ^t and ψ^t are three different 1×1 convolutional layers, w^a is the proposal appearance similarity measure function, which is defined the same as w^p in Eq. 2, M is the number of support proposals that have the same (predicted) class label with the target proposal, $\mathbf{Y}^t$ is the final aggregated feature for the target proposal, and $\mathcal{A}$ represents a feature alignment operation (this operation will be introduced in details in the following subsection III-E) to align the features of the support proposals to the ones of the target proposal.

2) Spatial Feature Aggregation (SFA): It has been well believed in computer vision community that relations between objects can help object recognition [12], [46], [47], [48].

Therefore, we introduce a spatial relation module to further explore the spatial topology relation of objects by embedding the additional position and shape information of the proposals besides its appearance feature to facilitate the video object detection.

The SFA shares the same architecture with the CPFA and CTFA (i.e., multi-head and multi-layer). However, the Spatial Relation head (SR head) in SFA is an extension of the TA head in CTFA. Besides capturing the appearance similarity between a proposal pair with an appearance similarity weight w^a as we do in the CTFA, a geometric weight w^g is also calculated to capture the topology relation between object proposals in the same target frame by using the shape and location information of the proposals:

$$w^g(\varphi(\mathbf{g}^t), \varphi(\mathbf{g}^s)) = \varphi(\mathbf{g}^t) \odot \varphi(\mathbf{g}^s), \tag{5}$$

where φ is a position embedding operation, $\mathbf{g}^t$ and $\mathbf{g}^s$ are the geometry information of the target proposal and support proposal, respectively, which are defined as

$$\begin{aligned} \mathbf{g}^t &= (x^t, y^t, h^t, w^t) \\ \mathbf{g}^s &= (x^s, y^s, h^s, w^s), \end{aligned} \tag{6}$$

where x^t and y^t are the location of the target proposal bounding box center, h^t and w^t represent the height and width of the target proposal bounding box, respectively. Symbols mean the same for support proposal i. The location and shape information of the proposal bounding box can be obtained from the RPN. The new geometric weight w^g is designed to model the topology relation of objects and only consider the relative geometric relationship between objects, which can guarantee that the aggregation is invariant to scale transformation.

The final similarity w between the target proposal p^t and the support proposal p^s is computed by combining the geometric similarity w^g with the original appearance similarity weight w^a

$$w = \frac{w^g \cdot exp(w^a)}{\sum_i w^g \cdot exp(w^a)} \tag{7}$$

which enables the SR head in the SFA to capture both the topology relation of objects and the appearance similarity of objects in the same frame. Note that the spatial feature aggregation is only performed in the target frame after its temporal aggregation.

E. Feature Alignment Module (FAM)

When performing feature aggregation for the target proposal with features of support proposals, it is highly possible that the objects in the target proposal and support proposal have quite different poses, shapes, etc., which makes the appearance features of these two proposals misaligned, and further degrades the feature aggregation. Therefore, appearance feature alignment is crucial for better feature aggregation. Different from FGFA [7] and MANet [8] which adopt FlowNet [41] to align features, and STMM [38] which adopts a local "MatchTrans" for feature alignment, we design a Feature Alignment Module (FAM) which tries to align the support proposal feature to the target one globally.

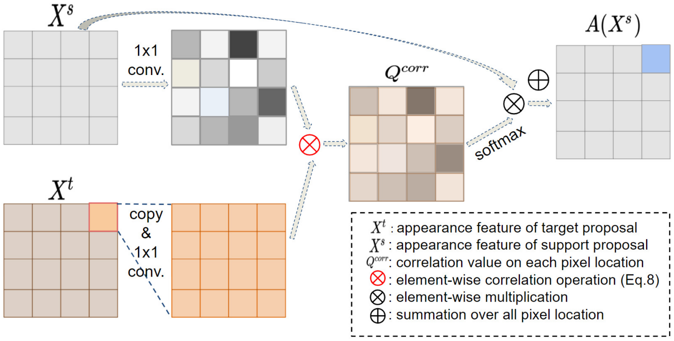
X^s
1x1
conv.
Q^{corr}
$A(X^s)$
softmax
X^t
copy
&
1x1
conv.
X^t : appearance feature of target proposal
X^s : appearance feature of support proposal
Q^{corr}: correlation value on each pixel location
⊗ : element-wise correlation operation (Eq.8)
⊗ : element-wise multiplication
⊕ : summation over all pixel location

Fig. 4: Idea of the proposed feature alignment module.

Fig. 4 depicts the idea of the designed feature alignment module. For better illustration, we only show the feature alignment at one pixel location in this figure. For the pixel location (u, v) in target proposal $\mathbf{X}^t$, a duplication operation is performed on its feature to generate a feature map which has the same size with the original target proposal feature map. Then, the duplicated feature map together with the feature map $\mathbf{X}^s$ of the support proposal goes through a 1×1 convolution layer to generate the target proposal relation feature $\mathbf{X}_r^t$ and support proposal relation feature $\mathbf{X}_r^s$. After that, a correlation map is calculated with these two relation features:

$$\begin{aligned} &Q_{t,s}^{corr}(u, v, x, y) = \\ &\frac{\Big(\mathbf{X}_r^t(u, v) - \mu(\mathbf{X}_r^t(u, v))\Big)\Big(\mathbf{X}_r^s(x, y) - \mu(\mathbf{X}_r^s(x, y))\Big)}{\sigma(\mathbf{X}_r^t(u, v)) \cdot \sigma(\mathbf{X}_r^s(x, y))} \end{aligned} \tag{8}$$

where $Q_{t,s}^{corr}(u, v, x, y)$ represents the correlation value between the support proposal feature at location (x, y) and the target proposal feature at location (u, v), $\mathbf{X}_r^t(u, v)$ denotes the relation feature vector at pixel location (u, v) of the target proposal, $\mu(\mathbf{X}_r^t(u, v))$ and $\sigma(\mathbf{X}_r^t(u, v))$ are the mean and standard variation of this feature vector, respectively, $\mathbf{X}_r^s(x, y)$ denotes the relation feature vector at pixel location (x, y) of the support proposal, $\mu(\mathbf{X}_r^s(x, y))$ and $\sigma(\mathbf{X}_r^s(x, y))$ are the mean and standard variation of this feature vector, respectively. After that, a softmax operation is applied on the correlation map along the spatial location dimension to obtain the alignment weight. Finally, the alignment weight Q^{corr} and the original support proposal feature map $\mathbf{X}^s$ are multiplied, so that the aligned feature at location (u, v) is obtained:

$$\mathcal{A}(\mathbf{X}^s)(u, v) = \sum_{x=1}^{U} \sum_{y=1}^{V} Q_{t,s}^{corr}(u, v, x, y) \cdot \mathbf{X}^s(x, y). \tag{9}$$

where U and V are the height and width of the proposal feature map, respectively.

IV. Experiments

The proposed framework is extensively evaluated in this section. First, the data and evaluation metric used in our experiments are introduced, followed by the detailed introduction of the network implementation. Next, ablation studies are conducted to evaluate the effectiveness of each proposed module. After that, the effect of sampling strategy of support frames on detection performance is studied and analyzed. Finally, the framework is compared with state of the art.

A. Dataset and Evaluation Metric

An intersection of the ImageNet DET and VID datasets [49] by taking their shared 30 object classes is used to train the proposed framework. The training and validation split settings in [7] are adopted here. The framework is evaluated on the VID validation set. The widely-used mean Average Precision (mAP)@IoU=0.5 is adopted as the evaluation metric.

B. Implementation Details

Backbone network The ResNet-101 [50] is adopted as the backbone network to extract features for each video frame.

Detection network The detection network is built upon Faster-RCNN. RPN [3] is applied on the feature extracted by $conv4$ of ResNet-101 to generate the object proposals for the target and support video frames. In total, 9 anchors with 3 different scales (i.e., 64^2, 128^2, 256^2) and 3 different aspect ratios (i.e., 1:2, 1:1, 2:1) are leveraged in RPN. During both training and inference, we first pick 6000 proposals with the highest objectness scores for each frame, then Non-Maximum Suppression (NMS) is performed on these proposals with IoU threshold of 0.7 to finally keep 300 proposals for each frame. RoIAlign pooling followed by a fully connected layer is applied on the $conv5$ feature to extract the feature for each proposal.

Training and testing The proposed model is trained end-to-end on 4 GPUs. We first initialize the backbone network with the pre-trained weights on ImageNet classification, then all modules in the model are trained and optimized simultaneously. Note that the RPN, object proposal classifier, CPFA, CTFA, SFA and the final detection layers are trained from scratches. A total of 10 epochs are performed to train the model with a SGD optimizer. Batch size is set to 4 with each GPU holds one minibatch. We use an initial learning rate of $2.5e^{-4}$, which is divided by 10 after 4 epochs, and divided again after another 4 epochs. During training, every training target frame is sampled along with two random support frames in the same video sequence (identical frames for the ImageNet DET dataset). When testing, for every inference frame (target frame), another F frames will be randomly sampled from the same video sequence as the support frames.

C. Comparison with State of the Art

To evaluate the effectiveness of our proposed model, we compare it with some state of the art, and summarize the results in Table I.

The comparison is first performed under the circumstance that all models are with the same backbone (ResNet-101). The results show that our model outperforms the single-frame object detection method D & T [9] (75.8% mAP) by a large margin (+9.2%). Besides, our model is remarkably better than FGFA [7] (76.3% mAP) and MANet [8] (78.1% mAP), which both aggregate features based on optical flow estimation, and the mAP improvements are +8.7% mAP and +6.9% mAP, respectively. When compared with some relation-based method (LRTRN [13] (81.0% mAP), RDN [39] (81.8% mAP), SELSA [14] (82.7% mAP)), MEGA [16] (82.9%mAP) and HVR-Net [18] (83.2%mAP), our method also shows its superior on detection precision. When a stronger backbone (ResNeXt-101) is adopted, better performance is achieved.

TABLE I: Comparison with state of the art on ImageNet VID validation set. 'X+Y' means post-processing strategy Y is employed on method X. IT denotes the inference time.

Method	Backbone	base detector	IT (ms)	mAP (%)
D & T [9]	ResNet-101	R-FCN	128.2	75.8
D & T + tubelet rescoring [9]	ResNet-101	R-FCN	-	79.8
FGFA [7]	ResNet-101	R-FCN	714.3	76.3
FGFA + Seq-NMS [7]	ResNet-101	R-FCN	-	78.4
MANet [8]	ResNet-101	R-FCN	200.0	78.1
MANet + Seq-NMS [8]	ResNet-101	R-FCN	-	80.3
ST-LA + tubelet rescoring [28]	ResNet-101	Faster R-CNN	50.0	79.6
STSN + Seq-NMS [10]	ResNet-101+DCN	R-FCN	-	80.4
STMN + Seq-NMS [38]	ResNet-101	Faster R-CNN	833.3	80.5
PSLA [11]	ResNet-101+DCN	R-FCN	-	80.0
PSLA + Seq-NMS [11]	ResNet-101+DCN	R-FCN	-	81.4
LRTRN [13]	ResNet-101	Faster R-CNN	100.0	81.0
RDN [15]	ResNet-101	Faster R-CNN	94.4	81.8
RDN + BLR [15]	ResNet-101	Faster R-CNN	-	83.8
SELSA [14]	ResNet-101	Faster R-CNN	820.3	82.7
SELSA + Seq-NMS [14]	ResNet-101	Faster R-CNN	-	82.7
MEGA [16]	ResNet-101	Faster R-CNN	238.1	82.9
MEGA [16]	ResNeXt-101	Faster R-CNN	-	84.1
MEGA + BLR [16]	ResNet-101	Faster R-CNN	-	84.5
LSTS [17]	ResNet-101	Faster R-CNN	43.5	77.2
LSTS [17]	ResNet-101+DCN	Faster R-CNN	47.2	80.1
LSTS + Seq-NMS [17]	ResNet-101+DCN	Faster R-CNN	217.4	82.1
HVR [18]	ResNet-101	Faster R-CNN	-	83.2
HVR + Seq-NMS [18]	ResNet-101	Faster R-CNN	-	83.8
HVR + Seq-NMS [18]	ResNeXt-101	Faster R-CNN	-	85.5
Ours	ResNet-101	Faster R-CNN	884.2	**85.0**
Ours	ResNeXt-101	Faster R-CNN	972.6	**86.1**

We then take the HVR-Net [18] as an example and analyze the reason of the performance gain of our proposed detection model. Compared with HVR-Net, our proposed CFA-Net has the following advantages: First, our proposed CFA-Net performs feature aggregation both in the pixel level and the proposal level, while the HVR-Net only conducts the proposal-level feature aggregation. For the proposal level feature aggregation, without accurate feature alignment, it is highly possible that a pixel in the target proposal is not aggregated with the most supportive ones, while the pixel level feature aggregation can characterize each pixel by exploiting its contextual information in both the target frame and support frames to enhance its feature representation. Second, we propose an additional class constraint for both the pixel level feature aggregation and the temporal proposal feature aggregation, i.e., a target pixel (or proposal) only aggregates its feature with the features of the support pixels (or proposals) that are in the same object class. The HVR-Net also performs the inter-video proposal feature aggregation, and there are more than one class objects in many videos. Without the class constraint, the relation mining mechanism in the feature aggregation module should be able to distinguish the same class support features from the different class features to select the most supportive features for feature aggregation, which intuitively increases the difficulties of the feature aggregation. Third, the HVR-Net exploits the intra-video and inter-video proposal relations, while ingores the intra-frame proposal relations which we leverage to boost video object detection by designing the Spatial Feature Aggregation (SFA) module in our proposed detection model. Last, when performing feature aggregation for target proposal with features of support proposals, it is highly possible that the objects in the target proposal and support

TABLE II: Ablation studies on the proposed modules. 'CPFA' is the class-aware pixel-level feature aggregation module, 'CTFA' is the class-aware temporal feature aggregation module, 'SFA' is the spatial feature aggregation module, and 'FAM' is the feature alignment module. mAP slow/medium/fast represent the detection precision for object with slow/medium/fast motion, respectively.

Method	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
CPFA		√				√		√
CTFA			√		√	√	√	√
SFA				√	√	√	√	√
FAM							√	√
mAP(%)	73.7	77.9	82.6	75.8	84.4	84.9	84.7	85.0
mAP(%)slow	82.4	85.3	88.1	84.4	88.9	89.4	89.2	89.4
mAP(%)medium	71.1	75.8	82.0	73.5	83.3	83.8	83.5	83.9
mAP(%)fast	51.7	56.7	67.6	54.1	69.1	70.2	69.7	70.4

proposal have quite different poses, shapes, etc., which makes the appearance features of these two proposals misaligned, and further degrades the feature aggregation. In our proposed CFA-Net, we design a Feature Alignment Module (FAM) which tries to align the support proposal feature to the target one globally when performing the temporal and spatial proposal feature aggregation, while the HVR-Net directly performs the proposal feature aggregation without any feature alignment.

D. Ablation Study on Proposed Modules

The effectiveness of each designed module in the CFA-Net is evaluated in this section, and the evaluation results are summarized in Table II.

(a) Baseline: This is the baseline detector without any feature aggregation, i.e., a single frame detector. It achieves a reasonable detection mAP of 73.7% as in [14].

(b) Effectiveness of CPFA: We add the class-aware pixel-level feature aggregation (CPFA) module into the baseline detector, and it achieves a +4.2% mAP improvement compared with the baseline. This is because pixel-level feature aggregation can enhance the pixel feature representation by encoding the context information from both target and support frames. However, the pixel-level feature aggregation is not able to effectively alleviate the partial occlusion problem, which is very common in video sequences.

(c) Effectiveness of CTFA: The class-aware temporal feature aggregation (CTFA) module is individually added into the baseline, which brings a +8.9%mAP improvement compared with the baseline. Also, compared with CPFA, the CTFA gives a better performance by robust occlusion handling. Besides, the CTFA gains larger improvement when the motion in video is faster, i,e., the gain for objects with fast motion is +15.9% (from 51.7% to 67.6%), while the gain for objects with slow motion and medium motion is +5.7% (from 82.4% to 88.1%) and +10.9% (from 71.1% to 82.0%). This indicates that the CTFA indeed does its job of exploiting temporal information across frames, especially in fast motion cases where objects are more likely to have partial occlusion and motion blur in neighboring frames, and then the temporal relation exploited by the CTFA can leverage information in other video frames (support frames) to greatly alleviate these challenges.

(d) Effectiveness of SFA: The spatial feature aggregation (SFA) only gains a +2.1% mAP improvement compared with the baseline, which is the smallest gain among the improvements brought by other kinds of feature aggregation. The reason is that the spatial instance-level feature aggregation only exploits the appearance and topology relation of proposals within the same target frame, which can not effectively deal with most of the VOD challenges such as occlusion, motion blur and rare pose in the target frame.

(e) Effectiveness of instance-level feature aggregation: The temporal and spatial instance-level feature aggregation (CTFA+SFA) further improves the CTFA-only method by additionally exploiting both the appearance and topology information of the proposals in the target frame. It is worth noting that both separating the instance-level feature aggregation in spatial and temporal dimension and the order of performing instance-level feature aggregation matters. When mixing the temporal and spatial instance-level feature aggregation together (i.e., instance-level feature aggregation is performed in one aggregation module by fairly treating proposals in the support frames and the target frame, and no topology relation between proposals in the target frame is exploited), it gives us a -0.8% mAP degradation. While if we separate the instance-level feature aggregation and adopt a different order, i.e., first spatial, then temporal, the performance decreases by 0.3% mAP compared with the proposed order. The reason is that the proposals from the temporal domain are more reliable and can enhance the feature for spatial relation exploration, since the same object can appear in different frames, while can not appear in the same frame.

(f) Effectiveness of pixel-level plus instance-level feature aggregation: Combining the pixel- and instance-level (i.e., CPFA+CTFA+SFA) feature aggregation achieves better performance compared with only performing pixel-level or instance-level feature aggregation. Because the two-level feature aggregation can not only perform fine-grained feature aggregation in pixel level by effectively exploiting the context information lying in the instances from both the current frame and the support frames, but also perform instance-level feature aggregation by separately aggregating the heterogeneous temporal and spatial information.

(g) Effectiveness of FAM: The feature alignment module (FAM) designed for the CTFA and SFA can benefit the instance-level feature aggregation performance by aligning the support proposal features to the target one globally. Compared with the CPFA, the FAM brings less improvement to the instance-level feature aggregation. The possible reason is that the CPFA already aggregates pixel features by aligning the pixels in both the spatial and temporal dimension.

(h) Effectiveness of the CFA-Net: The proposed CFA-Net combines all the proposed modules together, and achieves the best detection performance (85.0% mAP). Though for the case where the CPFA is included, the improvement brought by the FAM is very marginal, it is not completely replaceable by the CPFA (+0.1% mAP is obtained compared with the detection result of without FAM).

E. Analysis of Class-aware Pixel-level Feature Aggregation

The Class-aware Pixel-level Feature Aggregation (CPFA) is performed on the pixels of the extracted proposal feature tensor along both the spatial and the temporal dimension, i.e., the feature of a pixel in a target proposal feature tensor is aggregated with the same class pixel features both in the current frame and the selected support frames. To better understand the CPFA, we separate it in the temporal dimension and the spatial dimension, i.e., we perform the Class-aware Pixel-level Spatial Feature Aggregation (CPSFA) and Class-aware Pixel-level Temporal Feature Aggregation (CPTFA) separately. The experiment results are summarized in Table III.

(a) Baseline: This is the baseline detector without any feature aggregation, i.e., an image object detector (we use the Faster R-CNN in our experiment).

(b) CPSFA: The Class-aware Pixel-level Spatial Feature Aggregation (CPSFA) is added into the baseline. From Table III we can see that the CPSFA only brings limited detection improvement. The reason is that the CPSFA performs the feature aggregation for a pixel in a proposal feature tensor only with the pixels of the same class proposals in the current frame, while in the ImageNet VID, the video frames that are with only one object in a class dominate the dataset. Thus, for most frames, the CPSFA aggregates a pixel feature of an object only with the ones of this object itself, while additional information can not be provided for the objects in these frames.

(c) CPTFA: The Class-aware Pixel-level Temporal Feature Aggregation (CPTFA) is individually added into the baseline. Compared to the CPSFA, the CPTFA achieves a much bigger detection improvement. This is because the CPTFA can enhance the pixel feature representation by encoding the object information from other frames. This can greatly provide much more additional information for the proposal pixels need to be enhanced in the current frame.

(d) CPSFA + CPTFA: Finally, we add both the CPSFA and CPTFA into the baseline, and the detection performance is further boosted over the CPTFA alone. The reason is that in some video frames, there are more than one objects belonging to the same class, e.g., a group of sheep in the last row of Fig.5. In this case, the CPSFA can enhance the pixel feature of an object with the ones of the same class objects in this current frame. Note that 'CPSFA + CPTFA' is slightly different from the original CPFA in our work, as 'CPSFA + CPTFA' performs the spatial and temporal pixel-level feature aggregation separately, while the proposed CPFA module performs the pixel-level feature aggregation along the spatial and temporal dimension simultaneously. But if we compare column (d) in Table III with column (b) in Table II, we can see that both of them achieve almost the same detection performance.

TABLE III: Ablation studies on the Class-aware Pixel-level Feature Aggregation (CPFA) module. 'CPSFA' is the class-aware pixel-level spatial feature aggregation, and 'CPTFA' is the class-aware pixel-level temporal feature aggregation. mAP slow/medium/fast represent the detection precision for object with slow/medium/fast motion, respectively.

Aggregation	(a)	(b)	(c)	(d)
CPSFA		√		√
CPTFA			√	√
mAP(%)	73.7	74.2	77.7	78.0
mAP(%)slow	82.4	82.6	85.2	85.4
mAP(%)medium	71.1	71.7	75.6	75.9
mAP(%)fast	51.7	52.4	56.5	57.0

Pixel size	1	2	4	8
mAP(%)	77.9	78.3	79.5	83.1
mAP(%)slow	85.3	85.5	86.3	88.5
mAP(%)medium	75.8	76.3	77.7	82.6
mAP(%)fast	56.7	57.4	60.0	68.4

TABLE IV: Study on the pixel size in the Pixel-level Feature Aggregation (CPFA). mAP slow/medium/fast represent the detection precision for object with slow/medium/fast motion, respectively.

Further, we conduct a study on the pixel size in the pixel-level feature aggregation. The original extracted proposal feature is with the spatial size of 8×8, i.e., an object proposal is divided into 64 pixels (each pixel is with the unit spatial size of 1×1, and corresponds to a small patch in the original image), it is intuitively considered that the pixel can hardly contain structure information of the object in this object proposal. We first divide each proposal into 2×2 (i.e., 4 patches) non-overlapping patches and consider each patch as a pixel, in this case, each pixel is with the spatial size of 4×4. Then, each proposal is divided into 4×4 (i.e., 16 patches) non-overlapping patches and each patch is considered as a pixel, in this case, each pixel is with the spatial size of 2×2. Finally, we regard each proposal as a big pixel, and this pixel is with the spatial size of 8×8. By doing this, we construct pixels with various spatial sizes, and perform the Class-aware Pixel-level Feature Aggregation (CPFA) on pixels with various sizes separately. Note that only the CPFA is included into the baseline to perform the detection, and the instance-level feature aggregation is not included in this study. The experiment results are shown in Table IV, from which we can see that the detection performance rapidly improves with the increase of the pixel size. The reason is that when we use pixels with larger spatial size, each pixel can contain more object structure information and appearance feature. Thus, the partial occlusion problem can be overcome relatively easier. It is worth noting that when we regard the whole proposal as a big pixel and perform the pixel-level feature aggregation, it is actually the instance-level feature aggregation. However, the performance is slightly better than the Class-aware Temporal Feature Aggregation (CTFA) (column (c) in Table II). This is because the CPFA is performed along both the temporal dimension and the spatial dimension simultaneously, while the CTFA aggregates the proposal only along the temporal dimension. Besides, the performance of CPFA that regards the proposal as a big pixel is worse than the performance of 'CTFA+SFA' (column (e) in Table II), i.e., separately performing the instance-level feature aggregation in the temporal dimension and the spatial dimension. This also callbacks our claim that the temporal and spatial proposal feature is heterogeneous and we should perform the instance-level feature aggregation separately in the temporal dimension and the spatial dimension.

F. Analysis of Class Constraint

In our proposed framework, an object proposal classifier is designed to predict the class label of each generated proposal.

TABLE V: Analysis of class constraint.

class constraint	CPFA-C	CTFA-C	CPFA-C&CTFA-C	CPFA&CTFA
mAP(%)	84.8	84.3	84.0	85.0
mAP(%) slow	89.2	88.7	88.5	89.4
mAP(%) medium	83.6	83.1	82.7	83.9
mAP(%) fast	69.9	69.0	68.3	70.4

The pixel-level feature aggregation and the temporal instance-level feature aggregation are performed only among proposals with the same (predicted) class label, i.e., a class constraint is added on these two aggregations. We evaluate the effectiveness of the class constraint on these two aggregations by checking the detection performance of these two aggregations without the class constraint, and the results are shown in Table V.

If class constraint is deleted from the pixel-level feature aggregation ('CPFA-C' column in Table V), i.e., all pixels of the support frames are support pixels, which is exactly how the non-local network [19] works, and this leads to a performance degradation of -0.2% mAP. This shows that the CPFA filters out much noisy information from the large scope of background and different class objects, and enhances representations of target pixels with the pixels of the same class object proposals which only contain very limited background information. If we delete the class constraint from CTFA ('CTFA-C' column in Table V), i.e., the CTFA enhances the target proposal feature by considering all the support proposals, even support proposals that do not belong to the same class with the target one, the detection performance worsens by a -0.7% mAP. This verifies that without the class constraint, the features of some support proposals that are not in the same (predicted) class with the target proposal will also be partially used to aggregate the target proposal feature, which can pollute the target proposal feature and make it ambiguous, even the feature similarities are measured to guide the feature aggregation. When the class constraint is removed from both the CPFA and CTFA (the 'CPFA-C&CTFA-C' column in Table V), the detection result becomes worse, with a -1.0% mP degradation compared to the detection result with class constraint on both of these two feature aggregation modules. This illustrates that the class constraint benefits both of these two aggregations.

G. Analysis of Support Frame Sampling Settings

During training, every training target frame is sampled along with two random support frames in the same video sequence (for the ImageNet VID dataset, 2 different frames are randomly selected in the same video clip, while for the ImageNet DET dataset, the identical target frame is duplicated and used for support frames). During inference, for every target frame, another F frames will be randomly sampled from the same video sequence as the support frames (only the ImageNet VID validation dataset is used for inference). Thus, we use a fixed number of support frames during training, while the number of support frames F during inference is flexible. For inference, the number of support frames F is an important parameter, and sampling more support frames usually yields better results [38], [7]. In our experiments, we adopt the promising random sampling strategy [14] to sample support frames. For a target frame, the frames in the shuffled video sequence are randomly selected as support frames without considering the temporal order, i.e., both frames before and after the target frame can be selected. The influence of the number of support frames on detection accuracy is summarized in Table VI. The detection performance improves consistently by sampling more support frames. The reason is that with more support frames, more appearance information (e.g., shape, pose, etc.) and context information can be exploited by the aggregation modules to enhance the feature of the target proposals. Then, the performance saturates when enough support frames are used. The reason is that with more support samples offered, more appearance information (e.g., shape, pose, etc.) have been mined, and adding more support frames does not bring in extra information. Given the fact that more support frames means longer processing time, we set the number of support frames as 20 in our following experiments, for the trade-off between detection precision and time efficiency.

TABLE VI: Effect of support frame number F on detection precision.

# frames	2	6	10	14	20
mAP(%)	82.6	84.0	84.7	84.9	85.0
mAP(%) slow	88.0	88.7	89.2	89.4	89.4
mAP(%) medium	80.4	82.7	83.5	83.7	83.9
mAP(%) fast	65.0	68.4	69.9	70.2	70.4

Then we take a closer look at how support frame number affects the detection of objects with different motion speeds in videos. Table VI show that objects with fast motion gain the most improvement by using more support frames, while objects under slow motion gain the least. This is consistent with our intuition. Usually objects under fast motion have much more appearance variation, and are more easily occluded by some other objects during video capturing. Sampling more support frames can provide various and prolific supplementary appearance information for the target proposal objects with deteriorated appearance, and therefore the detection performance is improved with more support frames. On the other hand, objects with slow motion usually have much less appearance variation in a video sequence, therefore sampling more support frames can not provide much extra information.

Next, we evaluate the effectiveness of the random sampling strategy. First, we perform testing with 20 consecutive support frames (i.e., 10 consecutive frames before the target frame and 10 frames after), and the performance is shown in the second column of Table VII ('Con W/O NMS'). Then we adopt the Seq-NMS post-processing to refine the result, which is shown in the third column of Table VII ('Con W/ NMS'). The result of randomly sampling support frames is in the last column of Table VII. From this table we can see that consecutively sampling 20 support frames to perform feature aggregation while without any post-processing performs the worst among these three methods. This is because 20 consecutive frames capture a scene happening within $\sim$1 second, which means the object motion and appearance information are limited, especially for objects under slow motion. Moreover, for objects

TABLE VII: Effect of different sampling strategies on detection precision. 'Con W/O NMS' means sampling 20 consecutive support frames and without Seq-NMS, 'Con W/ NMS' means sampling 20 consecutive support frames and with Seq-NMS, 'Random' means sampling 20 support frames from the video randomly.

Sampling strategy	Con W/O NMS	Con W/ NMS	Random
mAP(%)	81.4	82.7	85.0

with fast motion, it is very possible that the object appears, disappears and re-appears in the video, but the 20 consecutive support frames are only a small portion of the whole sequence and only provide limited object information for the objects in the target frame, making the feature aggregation not ideal. When the Seq-NMS post-processing strategy is adopted, video level object information can be explored for object detection, and the detection result of the consecutive sampling can be improved (+1.6% mAP). Random sampling strategy achieves the best performance (+3.8 % mAP) over consecutive sampling (both with or without Seq-NMS), showing that the random sampling strategy can capture the object information from the whole sequence, and is more robust to deal with fast motion, sudden shot change that Seq-NMS suffers from.

H. Analysis of Computation Efficiency and Complexity

To better evaluate the computation efficiency and computational complexity of the proposed modules, we separately calculate the number of parameters and the Floating Point Operations (FLOPs) in each module of the detection model. Our proposed detection model mainly consists of the following modules: the backbone for frame feature extraction (ResNet-101 is adopted for backbone in our experiments), the Object Proposal Classifier (OPC), the Class-aware Pixel-level Feature Aggregation (CPFA) module, the Class-aware Temporal Feature Aggregation (CTFA) module, the Spatial Feature Aggregation (SFA) module and the final Detection Head (DH). Note that the Feature Alignment Module (FAM) is embedded in the CTFA module and SFA module to align the feature maps of the target proposal and the support proposals.

When calculating the FLOPs for each module, we follow the same experiment settings as in most of our experiments. Specifically, for each target frame, we randomly select 20 frames in the same video clip as the support ones, and 300 object proposals are generated for each frame. Table VIII summarizes the number of learnable parameters and the FLOPs for each module in our proposed detection model.

From Table VIII we can see that in the proposed detection model, the backbone network used for frame feature extraction has the most learnable parameters and the most FLOPs. This means that the backbone network costs the most memory size and has the highest computational complexity. Thus, it could be a good way to optimize the backbone network if we want to speed up the detection (e.g., replace the heavy ResNet-101 with MobileNet [31], [32]), and some previous works [29], [30] have already exploited this strategy to accelerate the detection. The Class-aware Temporal Feature Aggregation (CTFA) module have the second most learnable parameters and FLOPs, and the ablation studies (Table II) on each module also show that this module is the most important one among the proposed modules to improve the detection accuracy. For example, when CTFA is removed from the method (columns (a), (b) and (d) in Table II), the related mAPs are much lower than those columns with the CTFA module. Compared with the backbone and the CTFA modules, the learnable parameters and FLOPs in the Class-aware Pixel-level Feature Aggregation (CPFA) module are much less. In other words, the proposed CPFA module improves the detection accuracy by occupying very small memory size and consuming little computational resource. The Spatial Feature Aggregation (SFA) module has a large number of learnable parameters, almost the same with the CTFA module, however, the FLOPs in the SFA module is much less than the ones in the CTFA module. The reason is that the CTFA module performs the proposal feature aggregation for the target proposals with the support proposals in the 20 support frames, while the SFA module conducts the spatial proposal feature aggregation for the target proposals only with the proposals in the target frame. Finally, we separate the Feature Alignment Module (FAM) from the CTFA module and SFA module to check the computational complexity and computation efficiency of this module alone, and from Table VIII we can see that the FAM module also has a small number of learnable parameters and FLOPs.

I. Failure Case Analysis

We show some failure cases in Fig.5. The first row is an example of false classification in the whole video clip. There are two different object classes in this video clip, domestic cat and monkey. The proposed detection model classifies the monkey as a domestic cat in each frame of this video clip. The monkey in this video clip has a similar appearance feature with some yellow cats in some other video clips, and the proposed object proposal classifier wrongly labels the monkey proposal as a cat. Further, the temporal and spatial proposal feature aggregation modules aggregate the monkey feature with cat feature, and finally the detection head classifies the monkey as domestic cat. The reason of this failure case is that our proposed detection model lacks the capability of exploiting the inter-video proposal relations. Thus, combining the intra-video and inter-video proposal relations is a possible way to solve this problem, as the HVR-Net [18] does. The second row shows an example of temporally inconsistent detection. In a frame of this video clip, a background proposal (i.e., a blur house) is incorrectly labeled as a car, while the detection model detects the neighboring frames correctly. This is because our proposed detection model does not fully exploit the temporal consistency property of the video. The detection is performed in a frame-by-frame manner, and each target frame is individually detected, although some temporal information is leveraged by aggregating features from other frames. One possible solution for this problem is to leverage the tracking technique, e.g., performing the detection for a short frame sequence instead of a single frame at one time by generating object proposal triplet, as is done in [36]. The last row presents an example of duplicated detection. There are three sheep in each frame of this video clip, however, the

TABLE VIII: Computational complexity analysis for each module in the detection model.

module	backbone	OPC	CPFA	CTFA	SFA	DH	FAM
parameters ($\times 10^6$)	42.6	0.06	0.7	16.0	15.0	1.1	0.1
FLOPs ($\times 10^9$)	159.6	0.4	13.8	136.9	5.7	1.1×10^{-3}	1.7

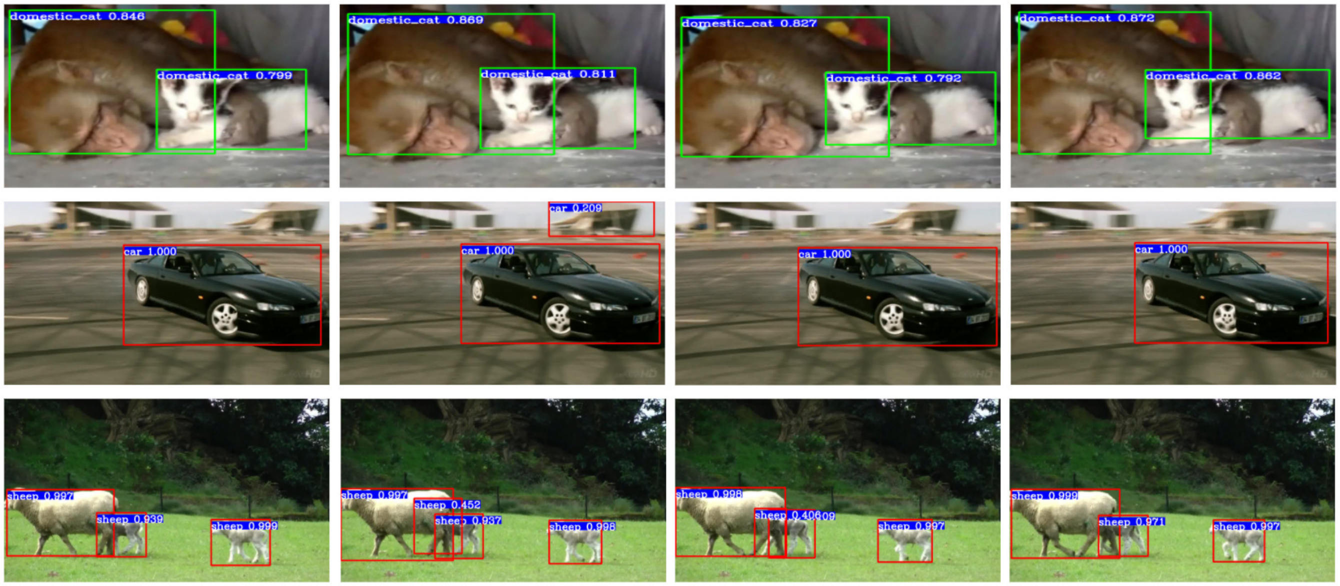

Fig. 5: Failure case analysis. First row: false classification in each frame of the video clip. Second row: temporally inconsistent detection. Third row: duplicated detection.

detection model detects four or even more sheep for some frames. Usually, the Non-Maximum Suppression (NMS) is adopted to delete the duplicated detection. Unfortunately, a pre-defined IOU threshold is used to perform the NMS for each frame, and this pre-defined IOU threshold can not work well for all the cases. Replacing the NMS operation by some specifically designed module might be helpful for this failure case, such as the relation network in [12].

V. Conclusion

In this work, we propose a class-aware feature aggregation network for video object detection. The class-aware pixel-level feature aggregation encodes each pixel with the context information from the same class instances, filtering out massive ambiguous information and enhancing the fine-grained feature representation. The class-aware temporal feature aggregation module considers the long-range temporal dependencies between objects in the same class across frames, and the spatial feature aggregation module exploits the topology relations between different objects in the same frame. The class-aware feature aggregation puts the video object detection to the edge, achieving state-of-the-art results.

References

[1] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[5] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, pp. 379–387, 2016.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

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

[8] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 542–557, 2018.

[9] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3038–3046, 2017.

[10] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 331–346, 2018.

[11] C. Guo, B. Fan, J. Gu, Q. Zhang, S. Xiang, V. Prinet, and C. Pan, "Progressive sparse local attention for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[12] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3588–3597, 2018.

[13] M. Shvets, W. Liu, and A. C. Berg, "Leveraging long-range temporal relationships between proposals for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9756–9764, 2019.

[14] H. Wu, Y. Chen, N. Wang, and Z. Zhang, "Sequence level semantics aggregation for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9217–9225, 2019.

[15] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Relation

distillation networks for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7023–7032, 2019.

[16] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory enhanced global-local aggregation for video object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10337–10346, 2020.

[17] Z. Jiang, Y. Liu, C. Yang, J. Liu, P. Gao, Q. Zhang, S. Xiang, and C. Pan, "Learning where to focus for efficient video object detection," in *European Conference on Computer Vision*, pp. 18–34, Springer, 2020.

[18] M. Han, Y. Wang, X. Chang, and Y. Qiao, "Mining inter-video proposal relations for video object detection," in *European Conference on Computer Vision*, pp. 431–446, Springer, 2020.

[19] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

[20] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 284–293, 2019.

[21] L. Han, P. Wang, Z. Yin, F. Wang, and H. Li, "Exploiting better feature aggregation for video object detection," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1469–1477, 2020.

[22] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[23] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[24] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[26] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2349–2358, 2017.

[27] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7210–7218, 2018.

[28] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. Change Loy, and D. Lin, "Optimizing video object detection via a scale-time lattice," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7814–7823, 2018.

[29] M. Liu and M. Zhu, "Mobile video object detection with temporally-aware feature maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5686–5695, 2018.

[30] M. Liu, M. Zhu, M. White, Y. Li, and D. Kalenichenko, "Looking fast and slow: Memory-guided mobile video object detection," *arXiv preprint arXiv:1903.10172*, 2019.

[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[33] C.-H. Yao, C. Fang, X. Shen, Y. Wan, and M.-H. Yang, "Video object detection via object-level temporal aggregation," in *European Conference on Computer Vision*, pp. 160–177, 2020.

[34] Z. Xu, E. Hrustic, and D. Vivet, "Centernet heatmap propagation for real-time video object detection," in *European Conference on Computer Vision*, pp. 220–234, 2020.

[35] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-nms for video object detection," *arXiv preprint arXiv:1602.08465*, 2016.

[36] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 817–825, 2016.

[37] 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," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, 2017.

[38] F. Xiao and Y. Jae Lee, "Video object detection with an aligned spatial-temporal memory," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 485–501, 2018.

[39] H. Deng, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, "Object guided external memory network for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6678–6687, 2019.

[40] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang, "Object detection in videos by high quality object linking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1272–1278, 2019.

[41] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.

[42] X. Chen, J. Yu, S. Kong, Z. Wu, and L. Wen, "Joint anchor-feature refinement for real-time accurate object detection in images and videos," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[44] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 244–253, 2019.

[45] J. Xu, Y. Cao, Z. Zhang, and H. Hu, "Spatial-temporal relation networks for multi-object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3988–3998, 2019.

[46] X. Chen and A. Gupta, "Spatial memory for context reasoning in object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4086–4096, 2017.

[47] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 891–898, 2014.

[48] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *2009 IEEE Conference on computer vision and Pattern Recognition*, pp. 1271–1278, IEEE, 2009.

[49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Liang Han received the B.S. and M.S. degree from Department of Mathematics, Shandong University, Shandong, China. He is currently working toward his Ph.D. degree at the Department of Computer Science, Stony Brook University, NY, USA. His current research interests include video object detection, biomedical image processing and analysis, multimodal price suggestion.

Pichao Wang received his PhD in computer science from University of Wollongong, Australia. He is now a senior algorithm engineer at Alibaba Group, USA. His current research interests include action recognition, video reID, video object detection, etc. Dr. Wang has published 60+ peerreviewed papers, including those in highly regarded journals and conferences such as IEEE TMM, IEEE THMS, CVPR, ICCV, AAAI, ACM MM, etc. He serves as the Area Chair of ICME 2021. He also serves as an Associate Editor of Computer Engineering from 2019.

Zhaozheng Yin is a SUNY Empire Innovation Associate Professor at Stony Brook University. He is affiliated with the AI Institute, Department of Biomedical Informatics, and Department of Computer Science. His group has been working on Biomedical Image Analysis, Computer Vision, and Machine Learning. Zhaozheng is an IEEE senior member and he served as Area Chairs for CVPR, ECCV, MICCAI, and WACV.

Fan Wang received the B.S. and M.S. degree from Department of Automation, Tsinghua University, Beijing, China, and the Ph.D. degree from Department of Electrical Engineering, Stanford University, California, United States. She is currently with Alibaba Group as a Staff Algorithm Engineer. Her research interests include object tracking and recognition, 3D vision and multi-sensor fusion.

Hao Li received his PhD in Optical Engineering from University of Chinese Academy of Sciences, China. He is now a senior staff algorithm engineer at Alibaba Group, China. His current research interests include network compression, face recognition, reID, image search, etc. Dr. Li has published 20+ peerreviewed papers and 20+ patents, including those in highly regarded journals and conferences such as CVPR, ECCV, ICCV, ICLR, etc.