Journal of Ambient Intelligence and Humanized Computing (2023) 14:497–509
https://doi.org/10.1007/s12652-021-03309-3

ORIGINAL RESEARCH

A feature temporal attention based interleaved network for fast video object detection

Yanni Yang[1] · Huansheng Song[1] · Shijie Sun[1] · Yan Chen[1] · Xinyao Tang[1] · Qin Shi[1]

Received: 27 October 2020 / Accepted: 1 May 2021 / Published online: 11 May 2021

Abstract

Object detection in videos is a fundamental technology for applications such as monitoring. Since video frames are treated as independent input images, static detectors ignore the temporal information of objects when detecting objects in videos, generating redundant calculations in the detection process. In this paper, based on the spatiotemporal continuity of video objects, we propose an attention-guided dynamic video object detection method for fast detection. We define two frame attributes as key frame and non-key frame, then extract complete or shallow features, respectively. Distinct from the fixed key frame strategy used in previous studies, by measuring the feature similarity between frames, we develop a new key frame decision method to adaptively determine the attributes of the current frame. For the extracted shallow features of non-key frames, semantic enhancement and feature temporal attention (FTA) based feature propagation are performed to generate high-level semantic features in the designed temporal attention based feature propagation module (TAFPM). Our method is evaluated on the ImageNet VID dataset. It runs at the speed of 21.53 fps, which is twice the speed of the base detector R-FCN. The mAP decline is only 0.2% compared to R-FCN. Effectively, the proposed method achieves comparable performance with the state-of-the-arts which focus on speed.

Keywords Object detection in videos · Self-attention · Feature propagation · Key frame · Feature similarity

1 Introduction

Object detection in videos is a critical and challenging research field in object detection, which has received increasing attention in recent years. With the rapid development of deep learning, the CNN-based detectors (Ren et al. 2015; Dai et al. 2016; Cai and Vasconcelos 2018; Zhang et al. 2020) have become the main-stream object detection algorithms. The state-of-the-art object detection methods have been demonstrated to show improved detection performance in accuracy. Since these existing studies mainly focus on detecting objects in single images, we define them as static detectors. However, compared to single images, videos include spatiotemporal information, that is, objects in video frames are continuous in temporal and spatial domains (Zhu et al. 2017b). Therefore, serious problems exist when using static detectors in video object detection. As shown in Fig. 1a, the features of consecutive frames in the video are similar. However, static detectors ignore the feature similarity and extract features for each frame, resulting in computational redundancy. On the other hand, as illustrated in Fig. 1b, video frames often suffer from the situations as object occlusion and motion blur, resulting in false positives. The state-of-the-art static detectors still cannot solve this accuracy degradation caused by the false positives. Therefore, the challenges of video object detection lie in computational redundancy and accuracy degradation bring by static detectors.

The key to solve the above challenges of computational redundancy and accuracy degradation is to exploit the spatiotemporal information of the videos. Feature propagation is an effective technique in video object detection. It is an application of spatiotemporal information at the feature level and can be used for feature association between frames. Feature aggregation is another important technique in video object detection, which is mainly used to improve the feature of some frames. In order to address the challenge of accuracy degradation, the video object detection methods of FGFA (Zhu et al. 2017a), STMN (Xiao and Jae Lee 2018),

✉ Huansheng Song
hshsong@chd.edu.cn

[1] Shool of Information Engineering, Chang'an University, Xi'an 710064, China

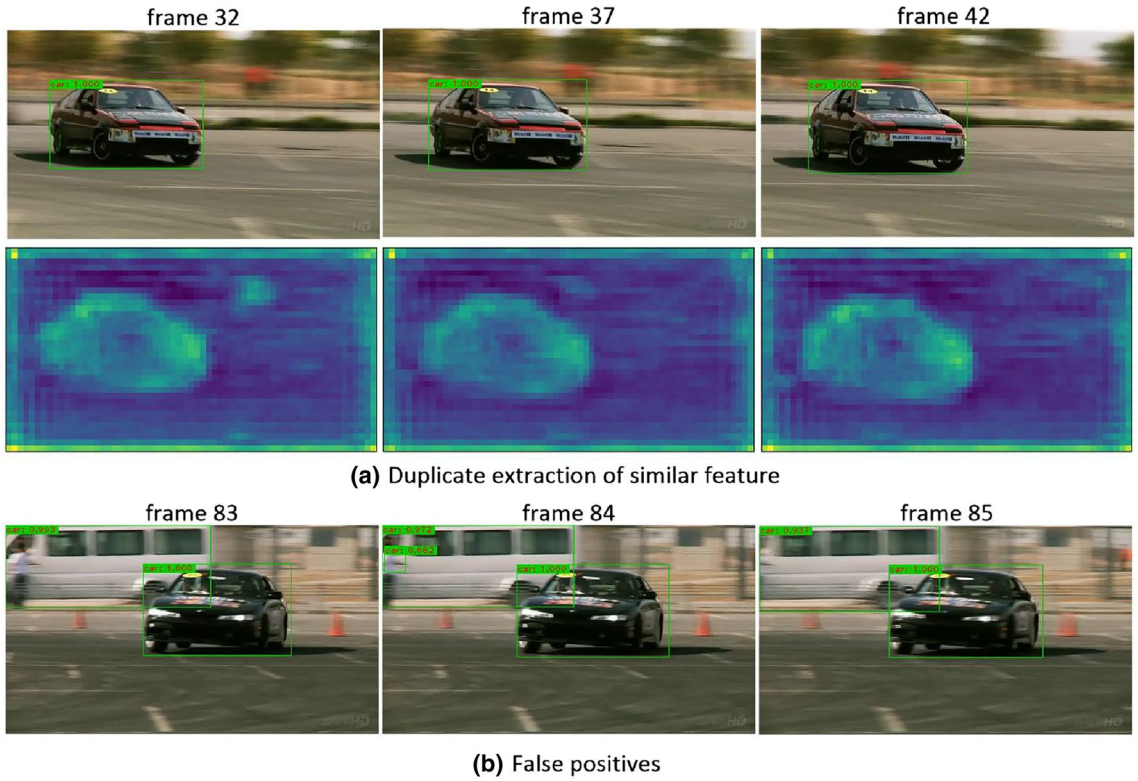

Fig. 1 Problems existing in video object detection caused by static detectors. **a** Duplicate extraction of similar features from neighboring frames. **b** False positives due to object blurring, occlusion, etc

and STSN (Bertasius et al. 2018) apply feature aggregation to strengthen the features of deteriorated frames using nearby frames. FGFA predicts pixel-level features of using optical flow (Dosovitskiy et al. 2015) and aggregates nearby features to improve the feature quality for each frame. Based on pixel-level feature calibration of FGFA, MANet (Wang et al. 2018a) fuses instance-level to deal with occlusion. However, these accuracy-focused studies enhance detection accuracy relying on expensive CNN-based feature extraction networks, thus, leading to low detection speed.

For the problem of high computing complexity existing in accuracy-focused studies, an ideal solution is to apply feature propagation to reduce computing cost while maintaining detection accuracy. In video object detection methods, optical flow using motion information and the memory based technology Long Short Term Memory Network (LSTM) are often used to propagate features. For instance, DFF (Zhu et al. 2017b) extracts feature maps for sparse key frames, and estimate feature maps for other non-key frames by optical flow. Because the complexity of optical flow network (Dosovitskiy et al. 2015) is lower than that of convolutional network, the total detection time is reduced. However, since the motion of high-level feature pixels is quite different from that of image pixels, estimating high-level features using optical flow representing image pixel motion may introduce artificial error. In addition, a fixed key frame strategy is used in DFF, resulting in missed detections for newcomers in non-key frames. Research of Liu et al. (2019) proposes an interleaved framework to propagate and aggregate features through LSTM (Xingjian et al. 2015). Moreover, an adaptive key frame policy using reinforcement learning (Hasselt et al. 2016) is used to further improve results. However, the inherent defect of LSTM is that object memory remains after it has moved to a different position, resulting in the inability of LSTM to accurately align features. In addition, LSTM is more time consuming. Thus it is not an optimal choice to propagate features with LSTM.

Self-attention mechanism (Vaswani et al. 2017) is a feature learning method more commonly used recently in vision analysis. For instance, attention mechanism (Bahdanau et al. 2015) is used for capturing long-range dependencies in Non-Local (Wang et al. 2018b), where connections between two pixels within an image or inter-frame are established using attention. Compared with optical flow and LSTM, self-attention mechanism directly calculates the correspondence between features, thus the attention-guided feature propagation method is more accurate and lightweight.

In this work, we focus on improving the video object detection speed by reducing the redundant computation in feature extraction while ensuring the detection accuracy.

Based on the high similarity between features of nearby frames and attention-guided feature propagation, we propose a dynamic video object detection network with a transformable feature extracting process. The complete and lightweight feature extracting networks are designed for sparse key frames and dense non-key frames, respectively. The extracted features of key frames are high-level semantic features, which are suitable to generate detection results. The low-level features produced by the lightweight feature extracting networks have fast extracting speed, but cannot be fed to the detection network. Thus feature propagation is employed to establish semantic features for non-key frames. In order to propagate features accurately and quickly, a reliable and lightweight feature propagation method named feature temporal attention (FTA) is introduced based on self-attention. We use the self-attention mechanism in time domain to establish connections between two feature pixels inter-frame. In addition, a lightweight transform network is used to further improve semantic information of low-level features. Based on FTA, the temporal attention based feature propagation module (TAFPM) predicts the final features of non-key frames by the key frame features and the transformed non-key frame features. Furthermore, in view of the fact that alternating frequency of complete and lightweight feature extracting networks is determined by key frame, we propose an adaptive key frame decision strategy using the similarity of low-level features from inter-frames. The integration of the TAFPM and key frame strategy leads to our video object network achieving comparable detection accuracy and greatly increases detection speed.

In summary, the contributions of this paper are as follows:

- We propose a new online dynamic video object detection network, which significantly improves detection speed by reducing redundant calculation in feature extraction.
- We introduce a lightweight attention-guided feature propagation method, which establishes an accurate connection between inter-frame features.
- We design a new adaptive key frame decision strategy based on the low-level features to further balance detection accuracy and computing time.
- We verify the proposed detection network on the ImageNet VID dataset, obtaining satisfactory detection performance.

2 Related work

2.1 Object detection in images

Currently, CNN-based approaches are the leading object detection methods. Since generating default boxes rely on anchors, methods in Ren et al. (2015), Dai et al. (2016), Liu et al. (2016), Bochkovskiy et al. (2020) and Cai and Vasconcelos (2018) are called anchor-based methods. In contrast, methods of CornerNet (Law and Deng 2020) and CentripetalNet (Dong et al. 2020) are anchor-free methods. Anchor-based methods fall in two categories: one-stage methods (Liu et al. 2016; Bochkovskiy et al. 2020) and two-stage methods (Ren et al. 2015; Dai et al. 2016; Cai and Vasconcelos 2018). YOLOv4 (Bochkovskiy et al. 2020)is a state-of-the-art one-stage method, which detects object by regression, and has a fast detection speed. However, compared with two-stage methods, the detection accuracy of one-stage methods is generally lower. Faster R-CNN (Ren et al. 2015) is the most representative two-stage method, which uses the idea of classification to detect objects. The extracted features are first used to propose possible regions, which are then classified to produce detection results. As a result, the Faster R-CNN is highly accurate, but time-consuming. R-FCN (Dai et al. 2016) increases the number of shared feature layers to 101, which generates an increase in the computing speed compared with Faster R-CNN. Cascade R-CNN (Cai and Vasconcelos 2018) combines the cascade idea and the Faster R-CNN detection framework, thus improving the detection accuracy. CornerNet uses the idea of keypoint to handle the object detection problem. It generates the detection box by finding the top-left point and bottom-right point. CentripetalNet is based on CornerNet. For the accurate match of keypoints, CentripetalNet proposes a corner matching method based on centripetal shift, along with a cross-star deformable convolutional module.

Based on the characteristics of the above detection methods, the two-stage method with higher accuracy is more suitable for our study. Therefore, R-FCN with ResNet-101 (He et al. 2016) is chosen as the static detector in the proposed video object detection method.

2.2 Object detection in videos

Video object detection methods incorporate video-specific spatiotemporal information into static detectors to improve the detection performance. The Spatiotemporal information can be fused in the post-processing stage or inside the static detection network. The former study is called the box level method, and the latter belongs to feature level method.

Box level methods operate on detection boxes in time domain in post-processing stage. For example, Seq-NMS (Han et al. 2016) propose sequence NMS, by which boxes of adjacent frames are linked to box sequences to boost weak detections. Seq-NMS can be embedded in other video object detection methods to further improve the detection performance. T-CNN (Kang et al. 2017b) utilizes box propagation to reduce false negatives, and introduces tracking to establish long-term connections of boxes. TCN (Kang et al. 2016) designs a strategy to classify and re-score tubelet.

D&T (Feichtenhofer et al. 2017) computes cross-correlation between features of adjacent frames to track the objects and forms tracklets, by which the inter-frame detections are linked to improve detection accuracy. By the proposed spatiotemporal cuboid proposal network, method in Tang et al. (2018) link detections in short and long range to improve the classification quality. These box-level methods use complex post-processing to enhance detection accuracy and become time-consuming.

State-of-the-art methods for detecting object in videos are feature level methods, where feature propagation and aggregation are usually applied to optimize detection structure. In the researches for boosting performance, it is a common operation to strengthen features by aggregating features from other frames, e.g., FGFA (Zhu et al. 2017a) and MANet (Wang et al. 2018a). The memory-guided method STMN (Xiao and Jae Lee 2018) aggregates feature by the proposed Spatial-Temporal Memory Module (STMM), and aligns feature with the MatchTrans module. STSN (Bertasius et al. 2018) uses deformable convolution to aggregate feature. Deng et al. (2019), Shvets et al. (2019) and Chen et al. (2020) aggregate features in the proposal-level. RDN (Deng et al. 2019) propagate and aggregate object relation over the supportive proposals. The aggregated features are then used to augment the feature of each reference object proposal. Shvets et al. (2019) proposes a temporal relation module to establish the similarities between inter-frame proposals and select proposals from nearby frame to strengthen the current proposals. In MEGA (Chen et al. 2020), the candidate box features of current frame are augmented by global and local information to achieve high accuracy. The above studies enhance detection accuracy at the cost of computing time.

Among the methods that consider speed and accuracy, (Zhu et al. 2018) combines the methods of DFF and FGFA, thus designs a common optical flow based detection framework for high detection performance. In addition, the proposed temporally-adaptive key frame scheduling also replaces the fixed key frame strategy in this work. TSSD-OTA (Chen et al. 2019) temporally integrates multi-scale features by ConvLSTM. Moreover, attention mechanism is introduced to selects optimal features for memory module ConvLSTM. Liu and Zhu (2018) proposes an efficient Bottleneck-LSTM to reduce computational cost in feature propagation. Later, Liu et al. (2019) designs a dynamic framework including multiple feature extractors and aggregates features using the Bottleneck-LSTM. Yao et al. (2020) integrates detection and tracking at the object level. The real-time tracker updates detections and propagates the box features between frames. LSTM is then used to aggregate the object-level features. Jiang et al. (2020) uses the idea of fixed key frame and propagates features by the proposed attention-based module of Learnable Spatio-Temporal Sampling (LSTS). From this collection of research of balancing detection speed and accuracy, feature propagation method and key frame strategy are key elements to reducing calculating speed while ensuring accuracy. For accurate and fast feature propagation, we use self-attention mechanism as relation module to model inter-frame dependencies on features.

2.3 Key frame strategy

Key frame idea is used to select sparse frames to improve computational efficiency when processing a video. It plays an important role in video object detection, video behavior recognition and video object segmentation. AdaFrame (Wu et al. 2019) proposes a framework to adaptively select relevant frames for fast video recognition. It uses a memory-agumented LSTM as the selector of the key frame. Li et al. (2018) and Xu et al. (2018) design lightweights CNN network to determine the key frames in video semantic segmentation.

In the existing research area of video object detection, most of the methods use fixed key frame strategies such as DFF, FGFA, MEGA, etc. Adaptive key frame strategy is adopted by methods in Liu et al. (2019) and Zhu et al. (2018). Zhu et al. (2018) defines key frame based on the output of optical flow. The density of key frame in Chen et al. (2018) depends on propagation difficulty. Relying on reinforcement learning, key frame is selected in Liu et al. (2019) and Yao et al. (2020). From the observation, the adaptive key frame has been less studied in video object detection. We propose an effective and lightweight key frame strategy by leveraging the feature itself, creating a complete and efficient detection network.

3 Methods

We develop an attention-based dynamic framework for video object detection, by which the running time is reduced while maintaining detection performance. In this section, we present the framework and implementation details. We first detail an overall outline of the framework. Then the principle of feature propagation and two component modules are introduced in detail: Feature temporal attention (FTA), temporal attention based feature propagation module (TAFPM), and key frame decision module.

3.1 Overview

Our method is based on the well-known static detector R-FCN. As shown in Fig. 2a, two steps are required to produce detection results in the network of R-FCN. Input images are first fed into CNN-based feature extractor N_f to produce feature maps f, which are then used as inputs of

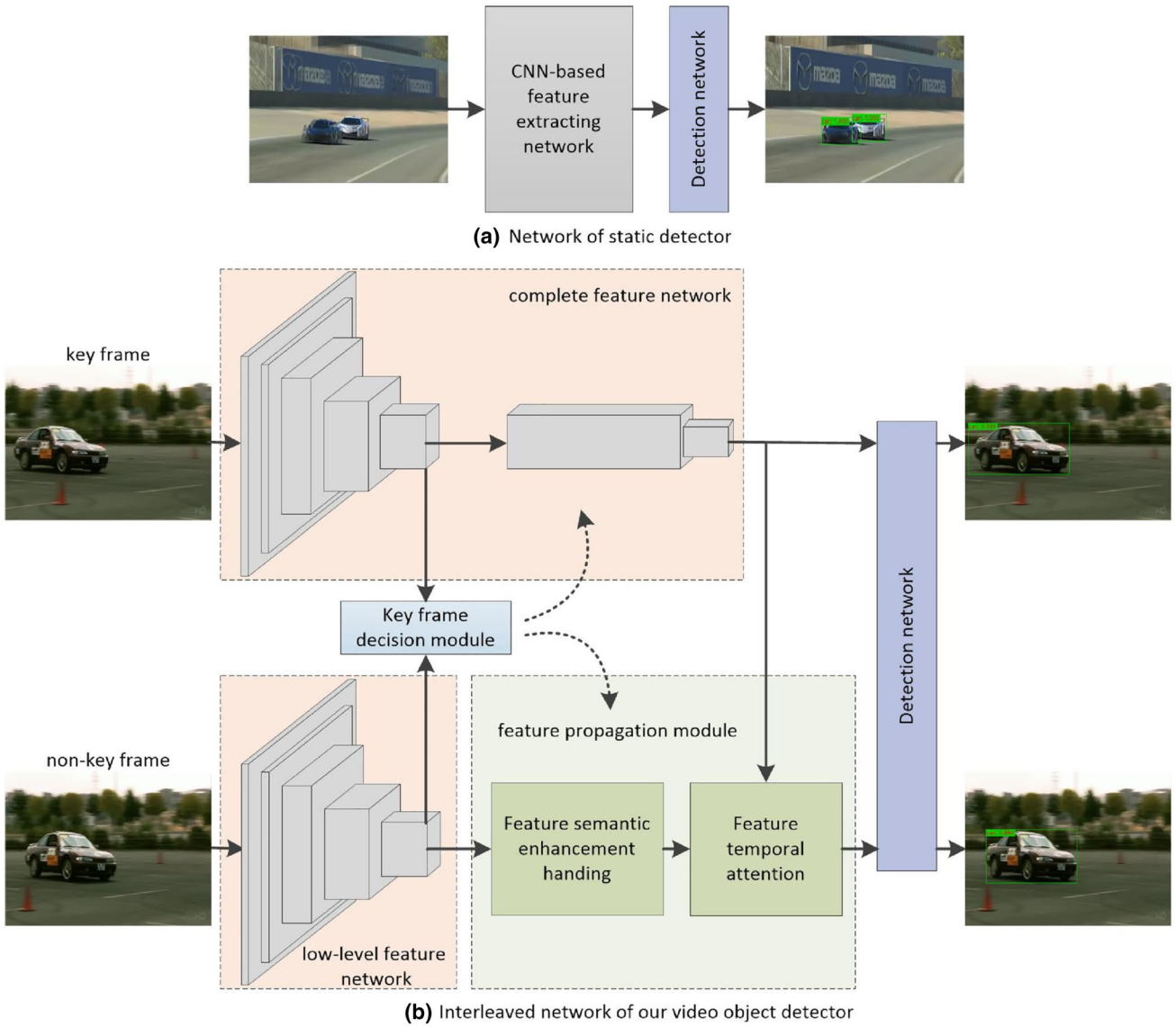

Fig. 2 Pipeline of the proposed video object detection method. Key frame decision module defines the properties (key or non-key) of each input frame. Key frame features are extracted via the complete feature network (ResNet-101). The lightweight low-level feature network and feature propagation module is designed for extracting and producing non-key frame features. Detection network is the same for each frame. It takes semantic features as input and outputs detection results

RPN to generate region proposals (RoIs). Finally, through position-sensitive RoI pooling layers and softmax layers, RoIs are processed to get the final detection results. Since undertaking the detection task, we define the subnetworks after N_f as detection network N_d. Compared with N_d, N_f is more time-consuming due to its multiple convolution operations. However, when a video sequence is served as the input, the output features of N_f are similar for neighboring frames, as shown in Fig. 1a. This means that extracting features for each frame is not necessary for video object detection, and the feature similarity of adjacent frames can be used to solve the computational redundancy in video object detection.

We propose a dynamic video object detection network to avoid the complex feature extraction for non-key frames. Moreover, based on the feature similarity, a key frame determination strategy is applied to further optimize the detection performance. Figure 2b illustrates the pipeline of the proposed dynamic framework. The key frames I_k and non-key frames I_t are defined by the key frame decision module, which is detailed in Fig. 4.

In this paper, we divide the original N_f into a low-level feature network N_f^l and high-level feature network N_f^h. Output features f^l of the lightweight N_f^l contain more detailed information. Semantic information of images needed in object detection is mainly reflected in the output features f^h of N_f^h. For key frames I_k, N_f is used to extract both low-level features f_k^l and high-level features f_k^h, that is, the final detection results of I_k are given by the complete R-FCN.

We define the starting frame of each input video as the first key frame. For each current frame, f_t^l is first extract by N_f^l. Then, f_k^l and f_t^l act as inputs for the key frame decision module to determine whether the current frame is the next key frame. If the current frame is a non-key frame, no high-level features are extracted. Given that the extracted low-level features f_t^l are less semantic for later detection tasks, feature semantic enhancement handling is designed to produce approximate high-level feature $f_t^{h_{appr}}$. Then the proposed feature temporal attention (FTA) acted on f_k^h and $f_t^{h_{appr}}$ to produce the propagated high-level feature of non-key frame f_t^h, which are followed by N_d to generate the detection results.

3.2 Feature temporal attention

Self-attention can assign weights to each feature unit through autonomous learning between feature maps, thereby extracting more useful feature maps. We use self-attention in the time domain, and propose feature temporal attention (FTA), by which high-level feature maps of key frames are propagated to non-key frames. FTA propagate feature through three steps. We first calculate the similarity between pairs of feature maps, and then normalize the similarity matrix to generate corresponding weights, based on which the propagated features are eventually produced.

We define the feature maps of frames I_k and $I_{k+\tau}$ as F_k and $F_{k+\tau}$, respectively, and both features have a size of $N * W * H$. The similarity matrix of the two feature maps is calculated by the dot-production function, as shown in Eq. (1):

$$f(F_k^i, F_{k+\tau}^j) = \theta(F_k^i)^T \phi(F_{k+\tau}^j) \tag{1}$$

where F_k^i represents an arbitrary position of F_k, similarly, $F_{k+\tau}^j$ corresponds to $F_{k+\tau}$. $f(\cdot)$ refers to the dot-production function, and the dimension of the output features is $WH * WH$. $\theta(F_k^i)$ and $\phi(F_{k+\tau}^j)$ are two embedding functions with the same processes. They are defined in Eq. (2):

$$\begin{cases} \theta(F_k^i) = W^\theta F_k^i \\ \phi(F_{k+\tau}^j) = W^\phi F_{k+\tau}^j \end{cases} \tag{2}$$

where W^θ and W^ϕ represent the same feature transformation for F_k^i and $F_{k+\tau}^j$, respectively. Taking W^θ as an example, first features F_k are convolved with the convolutional kernel of $(N/8) * 1 * 1$ to generate the intermediate features, which are then unfolded into a feature matrix with resolution of $(N/8) * WH$.

Since the similarity matrix $f(F_k^i, F_{k+\tau}^j)$ is used as a weight in self–attention mechanism, we normalize it with the softmax function to construct the attention map $att_{j,i}$ of F_k and $F_{k+\tau}$:

$$att_{j,i} = \frac{exp(f(F_k^i, F_{k+\tau}^j))}{\sum_{j=1}^{n} exp(f(F_k^i, F_{k+\tau}^j))} \tag{3}$$

where $att_{j,i}$ represent the attention to F_k^i when generating $F_{k+\tau}^j$. n indicates pixel number of $F_{k+\tau}$ after embedding, that is, all possible positions of j, $n = (N/8) * WH$.

According to attention map $att_{j,i}$ and F_k, the propagated feature map $F_{k+\tau}^{j_{pro}}$ of $I_{k+\tau}$ can be estimated with Eq. (4):

$$F_{k+\tau}^{j_{pro}} = \sum_{j=1}^{n} (att_{j,i} \cdot F_k^i) \tag{4}$$

where n is all possible positions of i, $n = N/8 * WH$. Through 1*1 convolution again, $F_{k+\tau}^{pro}$ is transformed into the same dimension as the extracted feature map $F_{k+\tau}$. $F_{k+\tau}^{pro}$ is represented by Eq. (5):

$$F_{k+\tau}^{pro} = (F_{k+\tau}^{1_{pro}}, F_{k+\tau}^{2_{pro}}, \ldots, F_{k+\tau}^{j_{pro}}, \ldots F_{k+\tau}^{m_{pro}}) \tag{5}$$

where m is the position number of $F_{k+\tau}^{pro}$, and which dimension is $N * W * H$. Therefore, through the rule of FTA, we propagate feature map of I_k to $I_{k+\tau}$.

3.3 Temporal attention based feature propagation module

Until this point, we have elaborated the basis (FTA) of feature propagation, that is how to update feature $F_{k+\tau}$ based on feature F_k. In FTA, the same-level features of two related frames are required. However, in our study, only low-level features are extracted in non-key frames for fast detection. Therefore, we propose a temporal attention based feature propagation module (TAFPM) to fix the feature propagation problem in this paper.

Next, we detail how to use high-level features of key frame f_k^h and low-level features of non-key frame f_t^l to obtain the high-level features of non-key frames. Since the high-level features used in subsequent detection network N_d express semantic information, which happens to be lacking in f_t^l, we design a lightweight network N_l for f_t^l to enhance semantic information. The resulting features are the approximate semantic features $f_t^{h_{appr}}$.

The structure of N_l is shown in Fig. 3. A convolution layer with a 1*1 kernel is first used to reduce the feature channels. In addition, the network includes two 3*3 convolutional layers with 512 and 1024 channels,

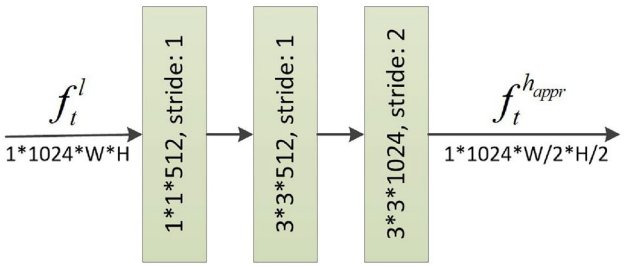

Fig. 3 The lightweight network for feature semantic enhancement handling. It takes low-level features (f_t^l) as input and outputs the estimated features $f_t^{h_{appr}}$ suitable for FTA

respectively. The output feature maps $f_t^{h_{appr}}$ have the same dimensions as f_k^h to ensure the implementation of feature propagation.

Feature propagation from key frame to non-key frame is performed based on FTA. We take f_k^h and $f_t^{h_{appr}}$ as inputs of FTA:

$$\begin{cases} F_k = f_k^h \\ F_{k+\tau} = f_t^{h_{appr}} \end{cases} \tag{6}$$

where f_k^h are the extracted high-level features of key frames, and $f_t^{h_{appr}}$ are the outputs of semantic enhancement handling of non-key frames.

After defining the inputs of FTA, through Eqs. (1)–(5), the output $f_t^{h_{pro}}$ are calculated as the propagated high-level features of non-key frame, which can be sent to N_d to produce the detection results of non-key frame.

3.4 Key frame decision module

The key frame module is a switching device for the proposed dynamic network. Feature similarity of video frames is the basis of the key frame module. Due to object emergence, disappearance, or change in appearance, feature maps of video frames will change with time. Research (Shelhamer et al. 2016) proves that, compared with semantic feature layers, intermediate layers can better reflect the changes in video frames. In this paper, we design an adaptive key frame decision method from the perspective of measuring low-level feature similarity.

As shown in Fig. 4, the module takes the previous key frame and current frame as inputs, and outputs feature similarity. The size of low-level features of previous key frame f_k^l and current frame f_t^l are defined as $N * W * H$. We first convolve f_k^l and f_t^l with 1*1*1 convolution kernel to reduce their feature channels to 1 , respectively. The resulting features are then unfolded into feature vectors with the size of $1 * WH$. The previous key frame and current frame feature vectors are denoted by v_k^l and v_t^l. Cosine similarity is used to calculate the similarity of these two feature vectors, so the similarity parameter $s_{k,t}$ of v_k^l and v_t^l is obtained by Eq. (7):

$$s_{k,t} = \frac{a_k^i \cdot a_t^i}{\left\|a_k^i\right\| \left\|a_t^i\right\|} \tag{7}$$

where a_k^i is an element of feature vector v_k^l, a_t^i belongs to feature vector v_t^l, $0 < i < W * H$. $\|\cdot\|$ denotes 2-norn.

Through $s_{k,t}$, the properties (key frame or non-key frame) of the current frame can be defined according to Eq. (8):

$$K_t = \begin{cases} 1, & s_{k,t} \geq \sigma \\ 0, & s_{k,t} < \sigma \end{cases} \tag{8}$$

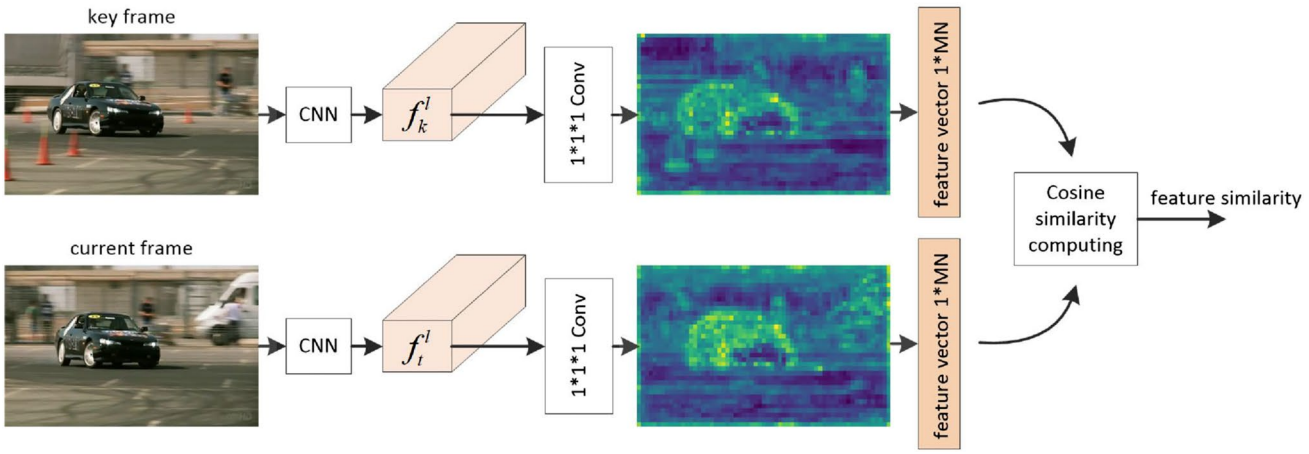

Fig. 4 Our key frame decision strategy. We process low-level features (f_t^l) to obtain feature similarity, which is used as the basis for determining key frames

where K_t stands for the indicator of key frames. The current frame t is a key frame when $K_t = 1$, otherwise, the value 0 means a non-key frame. σ represents the threshold of $s_{k,t}$, and $\sigma = 0.94$. The optimal value 0.94 is obtained by analyzing the influence of $s_{k,t}$ on accuracy and running time, which is detailed in the second part of the Experiment.

Figure 5 shows an example variation curve of similarity parameter $s_{k,t}$ with frame number. The first red point is the first frame of the video and is selected as the first key frame. The next six red points are the key frames selected by the proposed method. It is observed that frame difference of adjacent key frames is different. Also, non-key frames that farther away from the previous key frame has a smaller $s_{k,t}$.

4 Experiment

In this section, we evaluate our method on the ImageNet VID dataset, displaying the experimental results both qualitatively and quantitatively. All the experiments are on the computer equipped with a single GPU (NVIDIA GeForce GTX 1080 Ti) and 12 CPU (Intel i7-6800K), 32G RAM.

4.1 Experiment setup

4.1.1 Dataset and evaluation metric

The ImageNet VID dataset (Russakovsky et al. 2015) is the most preventative dataset for video object detection now. There are 5354 videos in the dataset, containing 3862 on training set, 555 on validation set, and 937 on testing set. The frames of training set and validation set are fully annotated. The 30 categories in VID dataset are a subset of the 200 categories in the DET dataset. The data of each category in VID dataset is imbalance. Additionally, sample quality of VID is poor than that of DET. Therefore, like most previous VID methods, we train detection model on the mixture of VID and DET (using the same category as VID). We sample 10 frames from each video in VID dataset and up to 2K images per class from DET dataset to compose our training set. As with the other video object detection research (Wang et al. 2018a; Zhu et al. 2017b), the detection performance is tested on the validation set.

Average precision (AP) and mean average precision (mAP) are the most widely used metrics in object detection. AP is defined as the mean precision corresponding to 11 recall values, which are produced by equally taken 10 points on the horizontal axis [0, 1] on the Precision-recall (PR) curve. We select AP and mAP to evaluate the accuracy of our method. Runtime is expressed in frames per second (fps). In experiments, following R-FCN, 0.5 is applied to the IoU threshold between RPN proposals and ground truth.

4.1.2 Implementation details and training

In our study, R-FCN is selected as the static detector. For feature extraction, we use ResNet-101 pre-trained on the ImageNet as our backbone network N_f. The convolution layer res4b3 is defined as the boundary between N_f^l and N_f^h. Layers up to res4b3 belong to N_f^l, and the higher layers are N_f^h.

The detection model is trained end-to-end with Stochastic Gradient Descent (SGD). During training, a sample consists of two frames, which are randomly sampled within a certain range in VID. The former acts as key frame and the latter is non-key frame. In order to produce samples with the same form as in VID, we replicate sampled images of DET once.

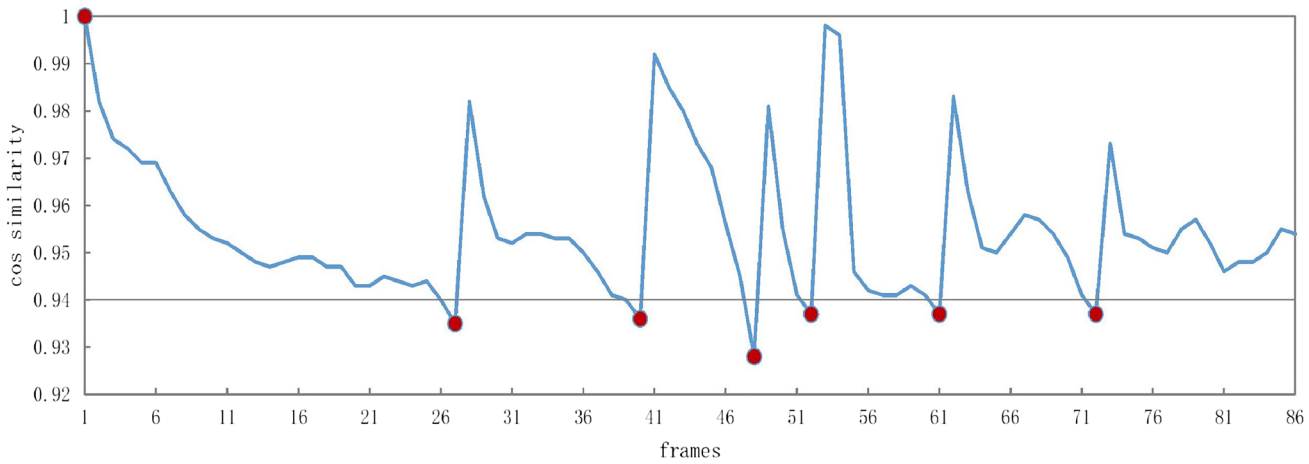

Fig. 5 Example variation of feature similarity. The red circles represent key frames. Between two key frames, feature similarity of key frame and current frame gradually decreases as the frame number increases

Iteration is set to be 120K, with learning rates of 10^{-3} and 10^{-4} in the first 80K and last 40K iterations, respectively. In both training and testing, input frames are resized such that their shorter side is 600 pixels.

4.2 Ablation study

4.2.1 Parameter analysis

Similarity threshold σ is an important parameter for key frame strategy. It determines the density of key frames and has a significant impact on detection accuracy and speed. We investigate the influence of σ on detection accuracy and running time, and display the results in Fig. 6. As σ rising, key frames become denser, detection accuracy increases, but runtime decreases. When σ takes the maximum value 1, each frame is a key frame and complete features are extracted. Thus the proposed detector is equal to the static detector. Conversely, lower σ produces sparse key frames, thus causes lower detection accuracy but faster running speed. Moreover, referring to the two curves, accuracy decreases slowly in initial stage, while the running time increases slowly in later stages. Since feature difference between frames is mainly caused by objects, the difference has a limit, which corresponds to the minimum value of feature similarity. When σ is too low, key frames are too sparse, and most frames detecting objects according to the inaccurate propagated features, resulting in a sharply drop in accuracy. Based on above analysis and tradeoffs of accuracy and speed, we eventually choose 0.94 as the optimal σ.

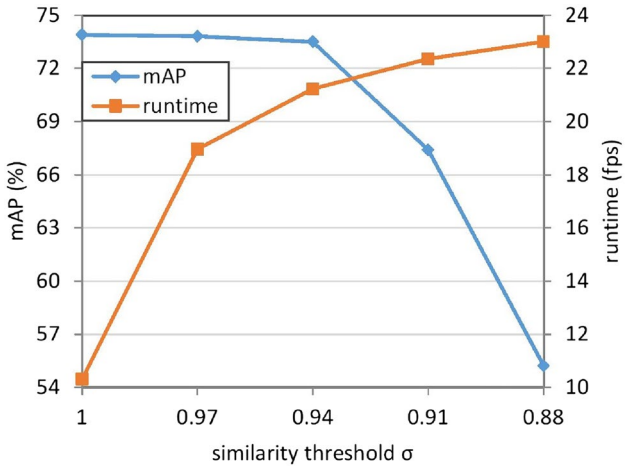

Fig. 6 Influence of feature similarity threshold σ on detection accuracy and running time

4.2.2 Tradeoffs of accuracy and speed

Table 1 shows the accuracy results of our method and the base detector R-FCN on the ImageNet VID dataset. We obtain an mAP of 73.7%, which is just 0.2% lower than the base detector, compared with 73.9% produced by R-FCN. This demonstrates that the proposed FTA based feature propagation method causes a slight decline in accuracy while accelerating processing speed. In addition, our AP is higher than R-FCN in several categories (e.g., bear, bus). The result is mainly due to the inter-frame association established at the feature-level. The extracted features are replaced with propagated features in non-key frames, thus avoiding detection failures on deteriorated non-key frames. This illustrates the necessity of inter-frame feature propagation in video object detection.

Since operation of detection network and post-processing are the same for each frame in our method, the total running time depends on the feature extracting time. Therefore, we analyze feature extracting times for both keys and non-key frames, as shown in Table 2. Input frames are preprocessed into 600*1000. Compared to 72 ms used in extracting complete features in key frames, it takes 12 ms extracting low-level features in non-key frames, which is five times lower than in key frames. The proposed key frame module and feature propagation module take 2 ms and 6 ms, respectively. Therefore, in non-key frame, we consume 20 ms to produce features available for object detection, which is less than 1/3 of key

Table 1 Average precision (in %) of our method and the base detector on the ImageNet VID dataset

Methods	Airplane	Antelope	Bear	Bicycle	Bird	Bus	Car	Cattle	Dog	Domestic	
R-FCN (Dai et al. 2016)	88.5	79.2	83.4	69.9	73.2	78.6	56.0	62.1	69.2	80.5	
Ours	88.1	79.0	**83.8**	69.5	73.1	**78.7**	55.9	61.8	68.7	80.1	
Methods	elephant	fox	giant panda	hamster	horse	lion	lizard	monkey	motorcycle	rabbit	
R-FCN (Dai et al. 2016)	77.1	86.6	79.8	87.8	73.7	49.2	77.5	51.4	79.9	66.2	
Ours	**77.4**	86.3	79.8	87.4	74.1	49.1	**77.7**	51.0	79.2	66.0	
Methods	Red panda	Sheep	Snake	Squirrel	Tiger	Train	Turtle	Watercraft	Whale	Zebra	mAP
R-FCN (Dai et al. 2016)	78.8	58.8	70.2	55.8	90.1	82.6	79.4	67.5	73.1	90.6	73.9
Ours	78.3	58.6	**70.5**	55.4	89.2	82.5	79.3	67.3	**73.9**	89.9	73.7

Table 2 Feature processing time (in ms) for key and non-key frames

Key frame	Non-key frame	Feature extraction	Key frame selecting	Feature propagation	Total
✓		72	2		74
	✓	12	2	6	**20**

Table 3 Performance comparison of fixed and adaptive key frame stretagy

Feature propagation	Fixed strategy ($L = 10$)	Adaptive strategy	mAP (%)	Runtime (fps)
✓	✓		73.3	20.72
✓		✓	**73.7**	**21.53**

frame. These results indicate that the proposed temporal attention based feature propagation module (TAFPM) demonstrates improvement in the speed of feature processing for non-key frames.

In order to verify the performance of our key frame strategy, we compare our strategy with the fixed key frame strategy. As shown in Table 3, when using the proposed adaptive key frame strategy, the mAP is 73.7%, which is 0.4% higher than using fixed key frame strategy. Similar results are achieved in terms of runtime, and our runtime is 0.81 fps higher than the fixed key frame strategy. This is because the fixed key frame strategy determines the properties of current frame based on frame difference, and the inter-frame feature variations are not taken into account. Therefore, large object appearance changes and emerging objects cannot be detected in time, resulting in a failure to detect the involved objects. Our adaptive key frame strategy effectively makes up for this deficiency. The enhancement in accuracy and speed proves that the proposed adaptive method is an improvement to the fixed key frame strategy.

4.2.3 Comparison with the state-of-the-art

Comparison with the state-of-the-art object detectors is reported in Table 4. Our method outperforms Faster R-CNN in both accuracy and runtime. We achieve 21.53 fps, which is about 3 times higher than Faster R-CNN. Unlike the comparable accuracy of our method and R-FCN, our processing speed is twice as fast as R-FCN. Among the compared video object detectors, due to using multi-frame feature aggregation to enhance feature quality, the accuracy-focus methods of FGFA, MANet, and STSN produce higher detection accuracy. However, the complex feature operations make detection speed of these detectors lower than that of DFF and TSSD-OTA. As a result, video object detectors (FGFA, MANet, and STSN) that focus on improving accuracy sacrifice speed for accuracy.

The optical flow based method DFF shares the same research focus and static detector as ours. Compared to the 73.1% mAP of DFF, we observe a 0.6% mAP improvement brought by the FTA and adaptive key-frame strategy. Our runtime is also 1.28 fps faster than DFF. The accuracy and runtime results prove than the proposed FTA based feature propagation method outperforms optical flow. In order to realize real-time processing, TSSD-OTA adopts a lightweight base network VGG16 (Simonyan and Zisserman 2015) and one-stage base detector SSD. TSSD-OTA runs at roughly the same speed as our method, but its mAP is 8.4% lower than ours. Since using the time-consuming Fast R-CNN (Girshick 2015) and LSTM, TPN has the lowest computing speed among these video object detection methods, and its mAP is 5.3% lower than the proposed method.

Figure 7 visualizes the qualitative detection results of the proposed method on the ImageNet VID validation dataset. We show six scenes. Scene 1 corresponds to the first row of images, and thus the sixth row is Scene 6. It can be seen from Scene 1 that the direction of the red car changes significantly (from an initial right front direction to a positive front direction, and finally a left front direction), our method successfully detects the car in all directions. In the remaining

Table 4 Accuracy and runtime comparison with state-of-the-arts on the ImageNet VID validation set

Methods	Base network	Base detector	Principle of feature propagation	mAP (%)	Runtime (fps)
Faster R-CNN (Ren et al. 2015)	ResNet-101	Faster R-CNN		73.4	5.61
R-FCN (Dai et al. 2016)	ResNet-101	R-FCN		73.9	10.31
FGFA (Zhu et al. 2017a)	ResNet-101	R-FCN	Optical flow	76.3	1.36
MANet (Wang et al. 2018a)	ResNet-101	R-FCN	Optical flow	78.1	4.96
STSN (Bertasius et al. 2018)	ResNet-101	Deformable R-FCN	Deformable convolution	**78.9**	
TPN (Kang et al. 2017a)	GoogLeNet	Fast R-CNN	LSTM	68.4	2.1
DFF (Zhu et al. 2017b)	ResNet-101	R-FCN	Optical flow	73.1	20.25
TSSD-OTA (Chen et al. 2019)	VGG-16	SSD	LSTM	65.4	21.00
Ours	ResNet-101	R-FCN	Attention	73.7	**21.53**

Fig. 7 Example detection results of our method on the ImageNet VID validation dataset. The images in each row belong to one scene. For each scene, we sample one frame every 5 frames and display its detection results. Our method achieves satisfactory results in these scenes

four scenes (Scene 3 with small objects and Scene 4 a complex scene), our method also detects objects accurately. For the case of large scale variation and occlusion in in Scene 6, our method is successful in detecting the car.

5 Conclusion

This paper aims at fast video object detection while ensuring detection accuracy. We propose an attention-guided dynamic video object detection method, by which complete and low-level features are extracted for the defined key frames and non-key frames, respectively. The complete features of key frames can be used for detection tasks. For non-key frames, the semantic information of low-level features is first enhanced through a lightweight network. Then, based on the proposed feature temporal attention (FTA), we propagate feature from key frames to non-key frames to produce the final features for detection. Furthermore, According to the feature similarity between frames, we design a new adaptive key frame decision method, which is served as the selection criteria for the two feature extraction processes. We demonstrate that our method offers a speed advantage while maintaining accuracy compared to the base detector. It is also competitive with the state-of-the-arts that focus on fast video object detection.

In the future, we will continue to study the algorithms of object detection in videos. We plan to further optimize the key frame decision method. The problem of key frame decision is to determine the interval between two adjacent key frames. Referring to the Keyframe Scheduling in Yao et al. (2020), the interval of key frames can be set to shorten interval, long interval, and mean interval, which correspond to fast change, slow change, and mean change of the objects, respectively. In this way the key frame decision problem can be viewed as a multiple attribute decision-making problem. Since the success of spherical fuzzy sets (SFSs) (Ashraf et al. 2019; Jin et al. 2019) and picture fuzzy sets

(PFSs) (Qiyas et al. 2020) in the field of decision-making, we will explore using the improved concept (e.g. linguistic picture fuzzy Dombi (LPFD) aggregation operators (Qiyas et al. 2019a) and Triangular picture fuzzy linguistic induced ordered weighted aggregation operators (Qiyas et al. 2019b)) to solve our key frame decision problem.

Acknowledgements This research was supported by the National Natural Science Foundation of China (62072053), the Fundamental Research Funds for the Central Universities (300102249317), Natural Science Foundation of Shaanxi Province (2019SF-258), and Key R & D project of Shaanxi Science and Technology Department (2019YFB1600500).

References

Ashraf S, Abdullah S, Aslam M, Qiyas M, Kutbi MA (2019) Spherical fuzzy sets and its representation of spherical fuzzy t-norms and t-conorms. J Intell Fuzzy Syst 36(6):6089–6102

Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015

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

Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:200410934

Cai Z, Vasconcelos N (2018) Cascade r-cnn: Delving into high quality object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6154–6162

Chen K, Wang J, Yang S, Zhang X, Xiong Y, Change Loy C, Lin D (2018) Optimizing video object detection via a scale-time lattice. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7814–7823

Chen X, Yu J, Wu Z (2019) Temporally identity-aware ssd with attentional lstm. IEEE Trans Cybern 50(6):2674–2686

Chen Y, Cao Y, Hu H, Wang L (2020) 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

Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387

Deng J, Pan Y, Yao T, Zhou W, Li H, Mei T (2019) Relation distillation networks for video object detection. In: European Conference on Computer Vision

Dong Z, Li G, Liao Y, Wang F, Ren P, Qian C (2020) Centripetalnet: Pursuing high-quality keypoint pairs for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 10516–10525

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

Feichtenhofer C, Pinz A, Zisserman A (2017) Detect to track and track to detect. In: Proceedings of the IEEE international conference on computer vision, pp 3038–3046

Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448

Han W, Khorrami P, Paine TL, Ramachandran P, Babaeizadeh M, Shi H, Li J, Yan S, Huang TS (2016) Seq-nms for video object detection. arXiv preprint arXiv:160208465

Hasselt Hv, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp 2094–2100

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Jiang Z, Liu Y, Yang C, Liu J, Gao P, Zhang Q, Xiang S, Pan C (2020) Learning where to focus for efficient video object detection. In: European Conference on Computer Vision

Jin H, Ashraf S, Abdullah S, Qiyas M, Zeng S (2019) Linguistic spherical fuzzy aggregation operators and their applications in multi-attribute decision making problems. Mathematics 7(5):413–434

Kang K, Ouyang W, Li H, Wang X (2016) Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 817–825

Kang K, Li H, Xiao T, Ouyang W, Yan J, Liu X, Wang X (2017a) Object detection in videos with tubelet proposal networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 727–735

Kang K, Li H, Yan J, Zeng X, Yang B, Xiao T, Zhang C, Wang Z, Wang R, Wang X et al (2017b) T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Trans Circuits Syst Video Technol 28(10):2896–2907

Law H, Deng J (2020) Cornernet: Detecting objects as paired keypoints. Int J Comput Vis 128(3):642–656

Li Y, Shi J, Lin D (2018) Low-latency video semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5997–6005

Liu M, Zhu M (2018) Mobile video object detection with temporally-aware feature maps. In: IEEE conference on computer vision and pattern recognition (CVPR)

Liu M, Zhu M, White M, Li Y, Kalenichenko D (2019) Looking fast and slow: Memory-guided mobile video object detection. arXiv preprint arXiv:190310172

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision, Springer, pp 21–37

Qiyas M, Abdullah S, Ashraf S, Abdullah L (2019a) Linguistic picture fuzzy dombi aggregation operators and their application in multiple attribute group decision making problem. Mathematics 7(8):764–785

Qiyas M, Abdullah S, Ashraf S, Khan S, Khan A (2019b) Triangular picture fuzzy linguistic induced ordered weighted aggregation operators and its application on decision making problems. Math Found Comput 2(3):183–201

Qiyas M, Abdullah S, Ashraf S, Aslam M (2020) Utilizing linguistic picture fuzzy aggregation operators for multiple-attribute decision-making problems. Int J Fuzzy Syst 22(1):310–320

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252

Shelhamer E, Rakelly K, Hoffman J, Darrell T (2016) Clockwork convnets for video semantic segmentation. In: European Conference on computer vision. Springer, pp 852–868

Shvets M, Liu W, Berg A (2019) Leveraging long-range temporal relationships between proposals for video object detection. In: IEEE international conference on computer vision, pp 9756–9764

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, pp 1–14

Tang P, Wang C, Wang X, Liu W, Zeng W, Wang J (2018) Object detection in videos by short and long range object linking. arXiv preprint arXiv:180109823

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008

Wang S, Zhou Y, Yan J, Deng Z (2018a) Fully motion-aware network for video object detection. In: Proceedings of the European conference on computer vision (ECCV), pp 542–557

Wang X, Girshick R, Gupta A, He K (2018b) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7794–7803

Wu Z, Xiong C, Ma CY, Socher R, Davis LS (2019) Adaframe: adaptive frame selection for fast video recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1278–1287

Xiao F, Jae Lee Y (2018) Video object detection with an aligned spatial-temporal memory. In: Proceedings of the European conference on computer vision (ECCV), pp 485–501

Xingjian S, Chen Z, Wang H, Yeung DY, Wong WK, Woo Wc (2015) Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810

Xu YS, Fu TJ, Yang HK, Lee CY (2018) Dynamic video segmentation network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6556–6565

Yao C, Fang C, Shen S, Wan Y, Yang M (2020) Video object detection via object-level temporal aggregation. In: European conference on computer vision, pp 160–177

Zhang W, Gao XZ, Yang CF, Jiang F, Chen ZY (2020) A object detection and tracking method for security in intelligence of unmanned surface vehicles. J Ambient Intell Hum Comput (2)

Zhu X, Wang Y, Dai J, Yuan L, Wei Y (2017a) Flow-guided feature aggregation for video object detection. In: Proceedings of the IEEE international conference on computer vision, pp 408–417

Zhu X, Xiong Y, Dai J, Yuan L, Wei Y (2017b) Deep feature flow for video recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2349–2358

Zhu X, Dai J, Yuan L, Wei Y (2018) Towards high performance video object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7210–7218