Temporal Context Enhanced Feature Aggregation for Video Object Detection

Fei He,[1,2] **Naiyu Gao,**[1,2] **Qiaozhe Li,**[1,2] **Senyao Du,**[3] **Xin Zhao,**[1,2] **Kaiqi Huang**[1,2,4]
[1]CRISE, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences [3]Horizon Robotics
[4]CAS Center for Excellence in Brain Science and Intelligence Technology
{hefei2018, gaonaiyu2017, liqiaozhe2015}@ia.ac.cn, senyao.du@horizon.ai, {xzhao, kaiqi.huang}@nlpr.ia.ac.cn

Abstract

Video object detection is a challenging task because of the presence of appearance deterioration in certain video frames. One typical solution is to aggregate neighboring features to enhance per-frame appearance features. However, such a method ignores the temporal relations between the aggregated frames, which is critical for improving video recognition accuracy. To handle the appearance deterioration problem, this paper proposes a temporal context enhanced network (TCENet) to exploit temporal context information by temporal aggregation for video object detection. To handle the displacement of the objects in videos, a novel DeformAlign module is proposed to align the spatial features from frame to frame. Instead of adopting a fixed-length window fusion strategy, a temporal stride predictor is proposed to adaptively select video frames for aggregation, which facilitates exploiting variable temporal information and requiring fewer video frames for aggregation to achieve better results. Our TCENet achieves state-of-the-art performance on the ImageNet VID dataset and has a faster runtime. Without bells-and-whistles, our TCENet achieves 80.3% mAP by only aggregating 3 frames.

Introduction

Object detection is a fundamental problem in computer vision. Deep convolutional neural networks have achieved remarkable results in this task, including (Ren et al. 2015; Redmon et al. 2016; Dai et al. 2016). Although they have been successfully applied to image-based object detection, video object detection remains a challenging problem. Object appearances in videos are usually deteriorated by motion blur or part occlusion, which are extremely difficult for image-based detectors.

To handle the object appearance deterioration problem, one straightforward solution is to consider the rich temporal and motion context in videos and leverage information from neighboring frames. Some methods (Kang et al. 2016; 2017; Han et al. 2016) exploit the video context in a post-processing manner, in which frame-based bounding boxes are firstly predicted by an image detector and then linked across time. However, these post-processing procedures cannot be unified into an end-to-end trainable framework. In contrast, some methods (Zhu et al. 2017a; Wang et al. 2018a; Bertasius, Torresani, and Shi 2018) attempt to exploit the video context to improve the per-frame feature by intuitive feature fusion. In these methods, fixed-length neighboring frames are used to enhance the appearance features at a reference frame, which is hereinafter referred to as spatial feature enhanced aggregation (SFEA) method. However, such SFEA approaches may ignore the temporal relations between the aggregated frames. Specifically, the performance of these methods is almost unaffected when shuffling the order of aggregated frames. It indicates that these models may not benefit from the temporal context modeling by ignoring the order of aggregated frames. By only performing appearance feature representation, it may be difficult to recognize some objects with severe appearance deterioration, such as rare poses, part occlusion.

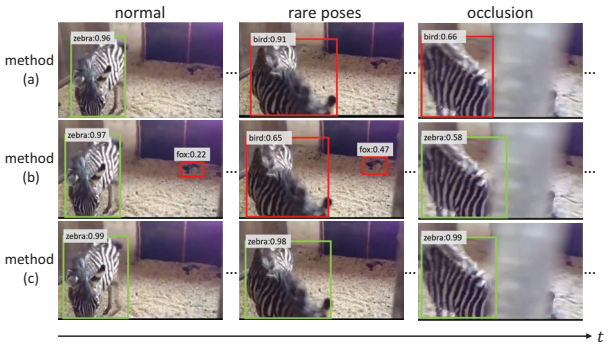

Figure 1: Examples of object appearance deterioration in video object detection. Method (a) is an image-based detector, method (b) is a detector employing the spatial feature enhanced aggregation method, method (c) is our proposed TCENet employing the temporal context enhanced aggregation method. When the zebra moves to a rare pose or is occluded, the image-based detector fails to obtain a correct box. Method (b) can help improve the results but it still performs inaccurate. TCENet performs the most accurate results among these methods.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper, our philosophy is that temporal context between neighboring frames play an important role in video object detection. Unlike traditional spatial feature enhanced aggregation methods, this paper proposes a temporal context enhanced aggregation (TCEA) method. It aggregates the features from neighboring frames to model the temporal context to enhance the features at a reference frame. Specifically, for each reference frame, the features from the neighboring frames, as well as its features on the reference frame, are aggregated according to attention weights and temporal order. Compared to the SFEA, the TCEA is more effective to handle severe appearance deterioration such as rare poses or occlusions. As is shown in Figure 1, when the zebra is occluded or in a rare pose, it still wouldn't be well recognized by the SFEA method. In comparison, our TCEA can significantly improve over the belief obtained from a single reference frame by taking the temporal context into account.

Furthermore, (Zhu et al. 2017a; Wang et al. 2018a; Bertasius, Torresani, and Shi 2018) aggregate long fixed-length video frames to obtain richer information, which makes it difficult to model variable temporal information and is computationally expensive. After taking a deeper look at the above aggregation, we find such densely feature aggregation may be inefficient because of the redundancy and dynamics of videos. Therefore, a temporal stride predictor is proposed to adaptively select video frames for aggregation. It assists TCENet to model variable temporal information and to reduce the number of aggregated frames. Besides, note that the features of the same object instance are usually not spatially aligned across frames due to video motion. To achieve more accurate pixel-level spatial alignment over time, a novel DeformAlign module is proposed to model the displacement introduced by motion across frames.

The main contributions are summarized as follows:

- A TCENet is proposed for video object detection which achieves state-of-the-art results on the ImageNet VID dataset.
- A TCEA method is proposed to model temporal context between aggregated frames, it is more effective to handle appearance deterioration.
- A temporal stride predictor is proposed to adaptively select video frames for aggregation, thus TCENet can exploit variable temporal information and requires fewer video frames for aggregation.
- A DeformAlign module is proposed to model the displacement introduced by motion across frames and achieve accurate pixel-level spatial alignment over time.

Related Work

Image-based Object Detection

Image-based detectors can be divided into two categories, two-stage detector, and one-stage detector. The pipeline of the two-stage detector can be summarized as generating region proposals first, then classifying and refining the proposals. Representative methods are R-CNN (Girshick et al. 2014), Fast R-CNN (Girshick 2015), Faster R-CNN (Ren et al. 2015). A two-stage detector is usually accurate but slow. In contrast, a one-stage detector is usually faster and simpler but less accurate. One-stage detector directly predicts the region proposals based on the feature map. Related works include YOLO (Redmon et al. 2016) and its variants (Redmon and Farhadi 2017; 2018), SSD (Liu et al. 2016), RetinaNet (Lin et al. 2017), FCOS (Tian et al. 2019). R-FCN (Dai et al. 2016) is an accurate and fast two-stage detector which proposes position-sensitive score maps and a position-sensitive RoI pooling layer. We use R-FCN as our base detector and extend it for video object detection.

Video Object Detection

Unlike static images, videos have richer information, and detectors for videos should take this information into account. Recent works on video object detection can be divided into two categories, object level, and feature level. The object-level works aim to explore the bounding box relations and apply temporal post-processing. And the feature level works are to leverage temporal coherence on features and try to do feature aggregation or feature propagation.

For object level, the main idea is to do box relation investigation or object-level detection-and-tracking. (Kang et al. 2016; 2017) propose to rescore tubelets. They apply a pretrained tracker to revisit the detection results and then associate image-based object detections around the tubelets. (Kang et al. 2016) proposes a re-scoring method to improve tubelets in terms of temporal consistency. (Kang et al. 2017) proposes multi-context suppression (MCS) to suppress false positive detections and motion-guided propagation (MGP) to recover false negatives. Seq-NMS (Han et al. 2016) proposes a cross-frame bounding boxes linkages using bounding box IoU and then rescores the boxes associated with each linkage to the average or maximum scores of the linkage. D&T (Feichtenhofer, Pinz, and Zisserman 2017) applies a bounding box tracker to predict object movements across frames while detecting. Then the detections are linked and re-weighted using the predicted movements. (Luo et al. 2019) uses detector and tracker on key frames and non-key frames respectively, to obtain detection results and track boxes. Key frames are selected by the key frame schedule network.

For feature level, DFF (Zhu et al. 2017b) utilizes the optical flow generated by FlowNet (Fischer et al. 2015) to estimate the per-pixel motion between two frames and align the features of selected key frames to neighboring non-key frames, reducing calculation and speeding up the system. FGFA (Zhu et al. 2017a) also applies optical flow to propagate features. The difference is that the propagated features are used for feature aggregation to enhance the features of reference frames to improve detection accuracy. STSN (Bertasius, Torresani, and Shi 2018) and FGFA have similar ideas, but the difference is that deformable convolution is used instead of the optical flow network. Based on FGFA, MANet (Wang et al. 2018a) adds an instance-level feature alignment and aggregation module besides the pixel-level feature alignment. Then these two-level features are combined through a motion pattern reasoning module. Different from previous works, STMN (Xiao and Lee 2018) applies a

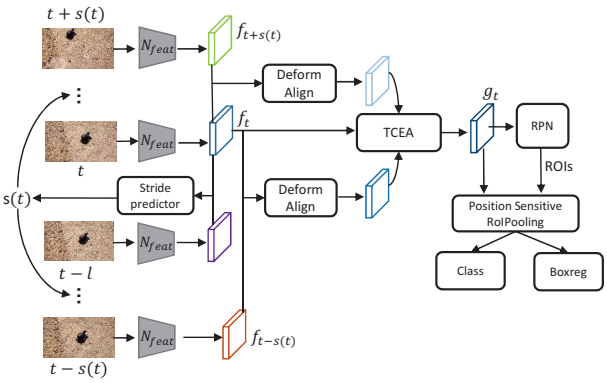

Figure 2: The framework of our TCENet.

MatchTrans module to align features. Then the aligned features are aggregate by the STMM module which utilizes recurrent computation unit.

Video Object Detection Framework

In this section, the entire pipeline of our framework is first briefly overview. Then, three key modules in our framework are introduced in turn, namely TCEA for figuring out how to aggregate the features from neighboring frames, DeformAlign for tackling object motion and aligning the features from frame to frame, and a temporal stride predictor for adaptively selecting video frames for aggregation. Finally, the inference and training process of our method is introduced in detail.

Framework Overview

The overall framework is shown in Figure 2. It is built on the image-based detector R-FCN (Dai et al. 2016). At each time step t, TCENet aggregates frames $t - s(t)$ and $t + s(t)$ with reference frame t, where $s(t)$ is calculated by temporal stride predictor. The feature extractor $\mathcal{N}_{\text{feat}}$ receives frames $I_{t-s(t)}, I_t$ and $I_{t+s(t)}$ as input, then produces the intermediate features $f_{t-s(t)}, f_t$ and $f_{t+s(t)}$. Prior to feature aggregation, the align module DeformAlign is applied to handle spatial feature mis-alignment between $f_{t-s(t)}, f_{t+s(t)}$ and f_t, generating $f_{t-s(t)\to t}, f_{t+s(t)\to t}$, which are then aggregated by our aggregation module TCEA to get g_t. Finally, the aggregated features g_t are fed to the detection network to obtain the detection results on the reference frame.

There are three modules in our framework: 1) TCEA. It figures out how to aggregate the features from neighboring frames. 2) DeformAlign. It tackles object motion and aligns the features from frame to frame. 3) Temporal Stride Predictor. It adaptively selects video frames for aggregation instead of fixed frames of aggregation. These key blocks of our model are elaborated below.

Temporal Context Enhanced Aggregation

Temporal Fusion. To model temporal context, the spatial feature enhanced aggregation approaches are insufficient as

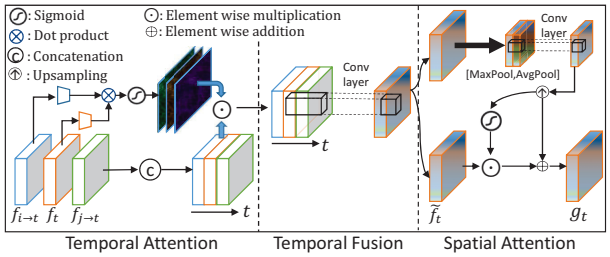

Figure 3: Temporal context enhanced aggregation module.

they may ignore the temporal relations between the aggregated frames. Specifically, the performance of these methods is almost unaffected by shuffling the order of aggregated frames, which indicates that their model may not benefit from the contextual relationships modeling between aggregated frames. As shown in the middle of figure 3, temporal fusion is proposed to aggregate features from neighboring frames to model temporal context. The features from N adjacent frames with size $C \times H \times W$ are firstly concatenated together to aggregate the temporal information, forming a NC-channel feature map. Unlike common appearance features, there is an additional temporal dimension. Then a convolution layer with $k \times k$ kernel is used to convolve with the concatenated feature map and capture temporal relations between frames. Finally, generate a C-channel feature map which both preserves temporal and spatial information. To reduce the number of parameters to be learned in the convolution kernel, k is set to 1. The experiment in ablation studies shows that our temporal fusion module is quite effective.

Temporal and Spatial Attention. Attention is proved to be effective in many tasks (Woo et al. 2018; Vaswani et al. 2017; Wang et al. 2019). Inspired by previous work (Zhu et al. 2017a) which indicates the importance of all neighboring frames to the reference frame at each spatial location by adaptive weight, attention modules are added to our TCEA to assign pixel-level aggregation weights on each frame. Specifically, temporal and spatial attentions are adopted as shown in Figure 3.

The goal of temporal attention is to compute frame similarity in an embedding space to focus on 'when' is important given neighboring frames. Intuitively, at location p, if the aligned features $f_{i\to t}(p)$ are close to the features $f_t(p)$, they should be paid more attention. Here, the dot product similarity metric (Wang et al. 2018b) is used to measure the similarity.

The weights of temporal attention map are estimated by:

$$M_t(p) = \sigma(f^e_{i\to t}(p) \cdot f^e_t(p)), \tag{1}$$

where σ is sigmoid function which restricts the outputs in [0,1], $f^e = \varepsilon(f)$ and $\varepsilon(\cdot)$ is an embedding network to reduce the features to 256 channels using convolution layer with 3×3 kernel. The temporal attention maps have the same spatial size with f_t and are then multiplied in a pixel-wise manner to the original aligned features $f_{i\to t}$.

Different from the temporal attention, the spatial attention focuses on 'where' is an informative part, which is com-

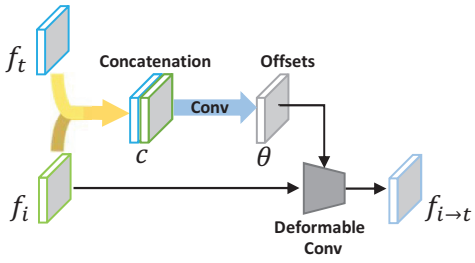

Figure 4: Architecture of DeformAlign module.

plementary to the temporal attention. Spatial attention maps are computed from the fused features generated by the temporal fusion module. To highlight informative regions and increase the attention receptive field, average-pooling and max-pooling operation are applied to the fused features first and two feature maps with half resolution are obtained. Then they are concatenated to generate a feature descriptor and a convolution layer is applied on the descriptor to generate an intermediate feature map. The intermediate feature map is upsampled with bilinear interpolation to generate a spatial attention map f_s. Similar to (Wang et al. 2018c), the spatial attention modulated features g_t is computed as:

$$g_t = \sigma(f_s) \odot \tilde{f}_t + f^{3\times3}(f_s), \tag{2}$$

where σ denotes the sigmoid function and $f^{3\times3}$ represents a convolution layer with 3×3 kernel. $\tilde{f}_t$ denotes the fused features and $\odot$ refers to element-wise multiplication. Through element-wise multiplication and addition, spatial attention provides fine-grained control to the features where should be emphasized or suppressed.

DeformAlign feature alignment

Note that the features of the same object instance are usually not spatially aligned across frames due to video motion. Without proper feature alignment before the aggregation, the object detector may obtain a lot of false recognitions and inaccurate localizations.

Therefore, the DeformAlign module is proposed to employ deformable convolution to achieve accurate pixel-level spatial alignment over time. The architecture of DeformAlign is shown in Figure 4. In order to transform the feature of frame i to align with that of reference frame t, the DeformAlign module first takes the f_i and f_t as inputs to predict sampling parameters Θ for the feature f_i:

$$\Theta = f_\theta(f_i, f_t) = \{\Delta p_n | n = 1, \cdots, |R|\}, \tag{3}$$

where $R = \{(-1,-1),(-1,0),\cdots,(0,1),(1,1)\}$ donates a regular grid of a 3×3 kernel. With Θ and f_i, the aligned feature $f_{i\to t}$ can be computed by the deformable convolution, for each position p_0 on the aligned feature map $f_{i\to t}$:

$$f_{i\to t}(p_0) = \sum_{p_n \in R} \omega(p_n) f_i(p_0 + p_n + \Delta p_n). \tag{4}$$

The convolution will be operated on the irregular positions $p_n + \Delta p_n$, where the Δp_n may be fractional. To address the issue, the operation is implemented by using bilinear interpolation, details can be found in (Dai et al. 2017).

For the sampling parameter generation function f_θ, f_i and f_t are first concatenated along axis 1. Then, the concatenated features are reduced to 256 channels using two convolution layers with 3×3 kernel. After that, there is a 1×1 kernel with $2 \times k \times k$ channels to generate offsets, where k is the kernel size of the deformable convolution. Finally, the aligned features $f_{i\to t}$ are obtained from offsets and f_i based on equation 4. Inspired from (Dollár, Welinder, and Perona 2010) and (Tian et al. 2018), an additional DeformAlign module is cascaded to further refine the coarsely aligned features.

Temporal Stride Predictor

To obtain richer information at a reference frame t, some models aggregate the long-term features of the input video frames based on a fixed-length sliding window. With a large convolution kernel, a large spatial receptive field can be obtained on the feature map. In the same way, expanding the length of the sliding window can increase the temporal receptive field to obtain more temporal information. However, the temporal neighborhood of the reference frame comprises mostly redundant information and is almost useless for improving the belief about the present object. Moreover, the large temporal length of the sliding window is computationally expensive.

Inspired by dilated convolutions (Yu and Koltun 2016), we find that increasing the temporal stride between aggregated frames can increase the temporal receptive field and aggregate more useful information without any computation increasing. Here the temporal stride s between two frames t_1 and t_2 in the same video is defined as $s = |t_2 - t_1|$. A naive temporal stride scheduling policy uses a fixed temporal stride at each reference frame t, e.g., aggregating frames $[t - s_0, t, t + s_0]$, which makes it difficult to model variable temporal information. A better temporal stride scheduling policy should be adaptive to the varying dynamics in the temporal domain.

A natural criterion for judging the temporal stride at a reference frame is the speed of the video content changes. If the speed is fast, choose a smaller temporal stride and aggregate the closer frames; on the contrary, choose a larger temporal stride and aggregate farther frames. The speed of the video content changes can be measured by the motion speed of the ground truth objects. An object's motion speed is measured by its intersection-over-union (IoU) scores with its corresponding instances in the neighboring frames (±10 frames). The indicator is dubbed as 'motion IoU'. The lower the motion IoU is, the faster the object moves.

Based on this, a temporal stride predictor is proposed for reference frame t to select which frames to aggregate. This predictor takes the differences between features t and features k, i.e. $(f_t - f_k)$, as input, and predicts the deviation score between frame t and frame k. The deviation score is formally defined as the motion IoU. If the predicted deviation is less than 0.7 (score < 0.7), the current reference frame sets a fast temporal stride (=9 by default). If the predicted score $\in [0.7, 0.9]$, the current reference frame sets a middle temporal stride (=24 by default). And the rest of

the situation (score > 0.9), the current reference frame sets a slow temporal stride (=38 by default). Specifically, this prediction network comprises two convolutional layers with 3×3 kernel and 256 channels, a global pooling, a fully-connected layer and a sigmoid function that follows. In runtime, at reference frame t, f_t and f_{t-10} are fed to this network to predict the motion speed of frame t.

Training and Inference

Inference. Algorithm 1 is a detailed summary of the inference algorithm. Given an input video of consecutive frames $\{I_i\}$, the specified aggregation range K (=1 by default) and the maximum temporal stride s_{max}, minimum temporal stride $s_{min}(s_{min} \geq 10)$. The proposed method sequentially processes each frame with a sliding feature buffer on the neighboring frames (of length $2Ks_{max}+1$ in general, except for the beginning and the ending Ks_{max} frames). At initial, the feature network is applied in the beginning $Ks_{max}+1$ frames to initialize the feature buffer and temporal stride (L3-L6 in Algorithm 1). Then the algorithm loops over all the video frames to perform video object detection, and to update the feature buffer. For each frame i as the reference, the aggregation 2K frames are sampled at stride s_i from the feature buffer, and the feature maps of the aggregate frames are aligned with respect to it (L10-L13). Then the aligned features are aggregated by our aggregation module to get aggregated feature g_i and fed to the detection network for object detection (L15-L16). Before taking the $(i+1)$-th frame as the reference, we calculate the temporal stride for frame $i+10$ with features of frame i and frame $i+10$ (L17). Finally, the feature maps are extracted on the $(i+Ks_{max}+1)$-th frame and are added to the feature buffer (L18-L19).

Training. The basic detection network, DeformAlign, and TCEA module are first trained following the settings in FGFA (Zhu et al. 2017a) and are then fixed as the feature extractor. After that, the temporal stride predictor is trained. It takes a pair of frames that are l steps apart as input(l is randomly chosen in [5,15]). Here, the motion IoU between the pair inputs is computed as the regression target based on the ground truth objects. If there are multiple objects, select the largest motion IoU.

Algorithm 1 Inference algorithm of temporal context enhanced feature aggregation for video object detection.

1: **input**: video frames $\{I_i\}$, aggregation range K, initialized temporal stride s_{min} and s_{max}
2: $F = [\,]$ ▷ feature buffer F
3: **for** $k = 1$ **to** $Ks_{max}+1$ **do** ▷ initialize F
4: $f_k = \mathcal{N}_{\text{feat}}(I_k)$
5: $F.append(f_k)$
6: $s_k = s_{min}$
7: **end for**
8: **for** $i = 1$ **to** ∞ **do** ▷ reference frame
9: $A = [\,]$ ▷ aggregate features buffer A
10: **for** $j = -K$ **to** K **do**
11: $n = \max(1, i + js_i)$
12: $f_{n\to i} = Align(f_n, f_i)$ ▷ align feature
13: $A.append(f_{n\to i})$
14: **end for**
15: $g_i = TCEA(A)$ ▷ aggregate features
16: $y_i = \mathcal{N}_{\text{det}}(g_i)$ ▷ detect on the reference frame
17: $s_{i+10} = Stride(f_i, f_{i+10})$ ▷ predict stride
18: $f_{i+Ks_{max+1}} = \mathcal{N}_{\text{feat}}(I_{i+Ks_{max}+1})$
19: $F.append(f_{i+Ks_{max+1}})$ ▷ update F
20: **end for**
21: **output**: detection results $\{y_i\}$

Experiments

Experiment Setup

Dataset. Following most of the previous video object detection works, we evaluate our method on the ImageNet (Deng et al. 2009) VID. VID dataset contains 3862 training videos and 555 validation videos. All videos are fully annotated with the object bounding box, object category, and tracking IDs. There are 30 object categories. They are a subset of the categories in the ImageNet DET dataset. Mean average precision (mAP) is used as the evaluation metric and all results on the validation set are reported following the previous methods (Zhu et al. 2017a; Lee et al. 2016).

Implementation Details. During training, following previous works, both the ImageNet DET training set and the ImageNet VID training set are utilized. Two-phase training is performed. In the first phase, the detection networks, the DeformAlign module, and TCEA are trained on ImageNet DET and ImageNet VID, only the same 30 categories are used. Each training batch contains three images. If they are sampled from DET, all images within the same mini-batch will be the same because DET only has images. If they are sampled from VID, two supporting frames are randomly sampled near the reference frame in the range of [-9,9]. In the second phase, the whole network except temporal stride predictor will be fixed. Then the predictor is trained based on the feature network with ImageNet VID. Each training batch has a pair of images, and the time step between them is randomly taken in [5,15]. In both training and inference, the images are resized to a shorter side of 600 pixels for the feature network.

Comparison to state-of-the-art

Table 1 shows the comparison of TCENet and other state-of-the-art methods, note all methods in the table use ResNet-101 (He et al. 2016) as the base network and R-FCN as the base detector. TCENet outperforms the image-based object detector R-FCN with a large margin (+6.6%), which demonstrates the effectiveness of our method. However, some methods use deformable R-FCN (DCN) (Dai et al. 2017) as the base detector, and others employ temporal post-processing techniques. To enable a fairer comparison to them, a TCENet with the deformable R-FCN based detector is also trained, and Seq-NMS (Han et al. 2016) is employed as temporal post-processing. TCENet achieves the best performance among various testing settings.

With R-FCN detector and no temporal post-processing.

Table 1: Comparison to the state-of-the-art methods on the ImageNet VID validation set.

Methods	Base network	Aggregate frames	Temp. Post-Proc	mAP(%)
TCENet (Ours)	ResNet-101	**3**		**80.3**
MANet (Wang et al. 2018a)	ResNet-101	13		78.1
FGFA (Zhu et al. 2017a)	ResNet-101	21		76.3
D&T (Feichtenhofer, Pinz, and Zisserman 2017)	ResNet-101	-		75.8
R-FCN (Dai et al. 2016)	ResNet-101	1		73.7
TCENet (Ours)	ResNet-101+DCN	**3**		**80.5**
STSN (Bertasius, Torresani, and Shi 2018)	ResNet-101+DCN	27		78.9
FGFA (Zhu et al. 2017a)	ResNet-101+DCN	21		78.8
Towards (Zhu et al. 2018)	ResNet-101+DCN	-		78.6
TCENet (Ours)	ResNet-101	**3**	✓	**81.0**
STMN (Xiao and Lee 2018)	ResNet-101	11	✓	80.5
STSN (Bertasius, Torresani, and Shi 2018)	ResNet-101+DCN	27	✓	80.4
MANet (Wang et al. 2018a)	ResNet-101	13	✓	80.3
D&T (Feichtenhofer, Pinz, and Zisserman 2017)	ResNet-101	-	✓	79.8
ST-Lattice (Chen et al. 2018)	ResNet-101	-	✓	79.6
FGFA (Zhu et al. 2017a)	ResNet-101	21	✓	78.4

Table 2: Accuracy and runtime of different methods on ImageNet VID validation. The runtime contains data processing which is measured on an NVIDIA Titan X Pascal GPU.

Methods	(a)	(b)	(c)	(d)	(e)
temporal fusion?		✓	✓		
DeformAlign?			✓	✓	✓
TCEA?				✓	✓
stride predictor?					✓
mAP(%)	73.7	$76.0_{\uparrow 2.3}$	$77.1_{\uparrow 3.4}$	$78.4_{\uparrow 4.7}$	$80.3_{\uparrow 6.6}$
runtime(ms)	81	82	120	124	125

Compared with MANet (78.1% mAP) and FGFA (76.3% mAP), TCENet obtains 80.3% mAP, outperforming these two methods by 2.2% and 4.0%. Furthermore, TCENet only aggregates 3 frames while these two methods are 13 and 21. It also outperforms D&T by a large margin of 4.5%.

With deformable R-FCN (DCN) detector and no temporal post-processing, STSN achieves the best performance of 78.9% mAP among all previous works. However, TCENet obtains 80.5% mAP, which is about 1.6% higher than it.

With temporal post-processing technique, TCENet still performs the most excellent mAP score of 81.0%. D&T and ST-Lattice adopt well-designed tubelet rescore technique and others use Seq-NMS (Han et al. 2016).

Ablation Study

TCENet Architecture Design. Table 2 compares TCENet with the image-based baseline and its variants.

Method (a) is the image-based baseline. It has a mAP 73.7% using R-FCN and ResNet-101, which is close to 73.4% mAP in FGFA (Zhu et al. 2017a). This indicates that our baseline is competitive and serves as a valid reference for evaluation. To verify the effectiveness of our method, we do not add bells and whistles like temporal post-processing, model ensemble, etc.

Method (b) only uses temporal fusion in Figure 3. It do not employ alignment module and temporal stride predictor. The variant is also trained in the same way as TCENet. After aggregating 3 frames, it increases the mAP score by 2.3% to 76.0%, with little increase in time.

Method (c) adds the feature alignment module which contains two cascading DeformAlign into (b). It obtains a mAP 77.1%, 1.1% higher than that of (b) and 3.4% higher than image-based detector R-FCN.

Method (d) uses completely TCEA which adds the temporal and spatial attention modules into (c). It increases the mAP score by 1.3% to 78.4%, which is a quite excellent performance.

Method (e) is the proposed temporal context enhanced feature aggregation method, which adds the temporal stride predictor to (d). It achieves a mAP 80.3%, 1.9% higher than that of (d). And in the case of image-based R-FCN, there is a 6.6% increase, which indicated the effectiveness of TCENet.

Aggregation Module. In this section, we seek to determine how much value the TCEA brings by replacing TCEA in our framework with other aggregation module. Here we choose the aggregation module which is widely used in previous work (Zhu et al. 2017a; Wang et al. 2018a; Bertasius, Torresani, and Shi 2018). We call it SFEA (Spatial Feature Enhanced Aggregation). For specific details, please refer to the article FGFA (Zhu et al. 2017a).

Theoretically, the SFEA module should be invariant to the temporal order of the input frames, since it is not capable of utilizing temporal relations between frames. To verify this, we train the SFEA model and TCENet with normal frame order, and employ the trained models on validation set in which the frames are in normal order, randomly shuffled temporal order and reversed order. The results of the experiment are shown in Table 3. Not surprisingly, the SFEA model has the same performance on all three versions during testing. In comparison, TCENet performs much worse on the randomly shuffled data than on the normal form of the data. However, its performance on the reversed form is the same, indicating that the model and/or dataset does not require inferring the causal 'arrow of time' (Pickup et al. 2014; Xie et al. 2018).

Table 3: mAP scores on ImageNet VID validation set. We train on frames in normal order and then test on frames in normal order, randomly shuffled order or reversed order.

Model	Normal(%)	Shuffled(%)	Reversed(%)
TCENet	80.3	14.3	80.3
SFEA	79.2	79.2	79.2

Table 4: mAP scores on ImageNet VID validation set. We train three models and use the same settings except the alignment module.

AlignModule	Flow-guided	STSN	DeformAlign
mAP(%)	78.8	78.9	79.8

Feature Alignment. To verify the efficiency of our DeformAlign module, we compare the DeformAlign module with the alignment module in the previous state-of-the-art method. Flow-guided alignment (Zhu et al. 2017a) and STSN (Bertasius, Torresani, and Shi 2018) are two alignment modules used in previous video object detection methods. To enable a fairer comparison, we follow the settings in STSN. We use deformable R-FCN as the base detector and SFEA as feature aggregation module, train three models to use three different alignment modules respectively, and test the performance on VID. Here, the temporal length of the sliding window is set to 27. The results are shown in Table 4. We can see the model with our DeformAlign module achieve a higher mAP score than the other two.

Temporal Stride. In this section, we conduct an ablation experiment to study the influence of the testing temporal stride. We do this by testing our TCENet model on different fixed temporal stride. For better analysis, besides the standard mAP scores, we also report the mAP scores over the fast, medium, and fast groups, respectively, denoted as mAP (fast), mAP (medium) and mAP (slow). The results are shown in Figure 5. Note that increasing the temporal stride does not bring about any increase in computation. With the increase of temporal stride, the mAP score gradually increases to a maximum point and then begins to decrease. This shows that increasing the temporal receptive field is very effective for improving the detection accuracy. However, the effective receptive field length is also limited and is distributed around the extreme points of the curve. We can see that the faster the object motion speed is, the smaller the effective receptive field is. Therefore, a temporal stride predictor can make sense.

Frames in Aggregation. The frames in aggregation are controlled by the aggregation range K in Algorithm 1. When the aggregation range is K, the number of aggregation frames is $2K + 1$. Due to the memory issues, we use a lightweight TCENet which abandons the attention module in the original TCEA and the stride predictor. We try 3 and 7 frames in aggregation and train two models. During the test, we adjust the temporal stride to make the temporal receptive fields of the two models consistent. The results are shown in Figure 6. We notice that when the temporal receptive fields are the same, fusing 7 frames can get higher

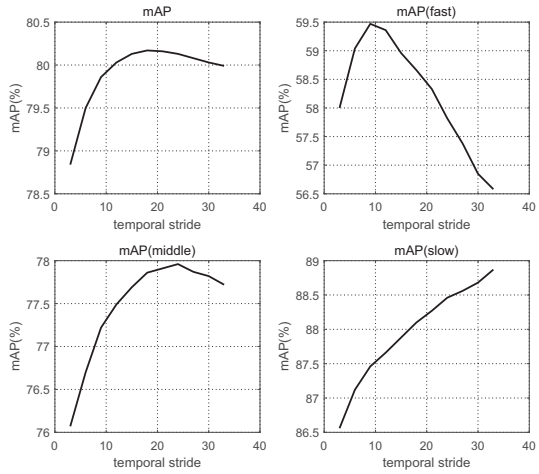

Figure 5: The influence of different temporal stride.

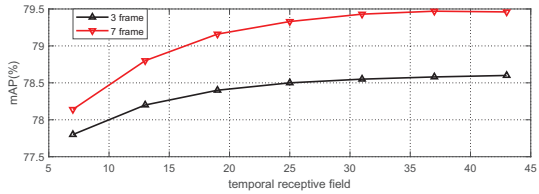

Figure 6: The influence of different frames in aggregation.

accuracy than 3 frames. However, the alignment module is evaluated $2K$ times for each frame. Therefore, in the feature alignment part, 7 frames require almost 3 times more computation than 3 frames.

Runtime. The performance and runtime of each component are listed in Table 2. TCENet takes 125ms to process one frame, using ResNet-101. As a comparison, FGFA (76.3 mAP) takes 256ms to process one frame and MANet (78.1 mAP) takes 202ms to process one frame. It is slower than the single-frame baseline (81ms) and most of the extra time is spent on the DeformAlign module. One reason is that the DeformAlign module is evaluated $2K$ (K is the aggregation range in Algorithm 1) times for each frame. Another reason is that deformable convolution in the DeformAlign module is slower than normal convolution. All the above results are tested on an NVIDIA Titan X Pascal GPU.

Conclusion

This paper proposes a temporal context enhanced feature aggregation framework to incorporate the temporal context for video object detection. Our main contributions are a feature aggregation module that models temporal context in features, a DeformAlign module that aligns the spatial features across time and a temporal stride predictor that adaptively selects video frames for aggregation. Ablation experiments show the effectiveness of our modules. Together, the proposed model achieves 80.3% mAP score on ImageNet VID dataset with backbone network ResNet-101, which achieves

state-of-the-art results with a competitive speed.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (Grant No. 2016YFB1001005), the National Natural Science Foundation of China (Grant No. 61673375, Grant No. 61602485 and Grant No. 61721004), and the Projects of Chinese Academy of Science (Grant No. QYZDB-SSW-JSC006).

References

Bertasius, G.; Torresani, L.; and Shi, J. 2018. Object detection in video with spatiotemporal sampling networks. In *ECCV*.

Chen, K.; Wang, J.; Yang, S.; Zhang, X.; Xiong, Y.; Loy, C. C.; and Lin, D. 2018. Optimizing video object detection via a scale-time lattice. In *CVPR*.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-FCN: Object detection via region-based fully convolutional networks. In *NeurIPS*.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *ICCV*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Li, F. F. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Dollár, P.; Welinder, P.; and Perona, P. 2010. Cascaded pose regression. In *CVPR*.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2017. Detect to track and track to detect. In *ICCV*.

Fischer, P.; Dosovitskiy, A.; Ilg, E.; Häusser, P.; Hazirbas, C.; Golkov, V.; van der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *ICCV*.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

Girshick, R. 2015. Fast R-CNN. In *ICCV*.

Han, W.; Khorrami, P.; Paine, T. L.; Ramachandran, P.; Babaeizadeh, M.; Shi, H.; Li, J.; Yan, S.; and Huang, T. 2016. Seq-NMS for video object detection. *arXiv:1602.08465*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Kang, K.; Ouyang, W.; Li, H.; and Wang, X. 2016. Object detection from video tubelets with convolutional neural networks. In *CVPR*.

Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; and Ouyang, W. 2017. T-CNN: Tubelets with convolutional neural networks for object detection from videos. *TCSVT*.

Lee, B.; Erdenee, E.; Jin, S.; Nam, M. Y.; and Rhee, P. K. 2016. Multi-class multi-object tracking using changing point detection. In *ECCV*.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2016. SSD: single shot multibox detector. In *ECCV*.

Luo, H.; Xie, W.; Wang, X.; and Zeng, W. 2019. Detect or track: Towards cost-effective video object detection/tracking. In *AAAI*.

Pickup, L. C.; Zheng, P.; Wei, D.; Shih, Y. C.; Zhang, C.; Zisserman, A.; Scholkopf, B.; and Freeman, W. T. 2014. Seeing the arrow of time. In *CVPR*.

Redmon, J., and Farhadi, A. 2017. YOLO9000: better, faster, stronger. In *CVPR*.

Redmon, J., and Farhadi, A. 2018. Yolov3: An incremental improvement. *arXiv:1804.02767*.

Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Tian, Y.; Zhang, Y.; Fu, Y.; and Xu, C. 2018. Tdan: Temporally deformable alignment network for video super-resolution. *arXiv:1812.02898*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully convolutional one-stage object detection. In *ICCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, S.; Zhou, Y.; Yan, J.; and Deng, Z. 2018a. Fully Motion-Aware network for video object detection. In *ECCV*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local neural networks. In *CVPR*.

Wang, X.; Yu, K.; Dong, C.; and Loy, C. C. 2018c. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*.

Wang, X.; Chan, K. C.; Yu, K.; Dong, C.; and Loy, C. C. 2019. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPRW*.

Woo, S.; Park, J.; Lee, J.; and Kweon, I. S. 2018. CBAM: convolutional block attention module. In *ECCV*.

Xiao, F., and Lee, Y. J. 2018. Video object detection with an aligned Spatial-Temporal memory. In *ECCV*.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning for video understanding. In *ECCV*.

Yu, F., and Koltun, V. 2016. Multi-scale context aggregation by dilated convolutions. In *ICLR*.

Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017a. Flow-Guided feature aggregation for video object detection. In *ICCV*.

Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017b. Deep feature flow for video recognition. In *CVPR*.

Zhu, X.; Dai, J.; Yuan, L.; and Wei, Y. 2018. Towards high performance video object detection. In *CVPR*.