TF-Blender: Temporal Feature Blender for Video Object Detection

Yiming Cui[1]*, Liqi Yan[2]*, Zhiwen Cao[3], Dongfang Liu[4]†
[1]University of Florida, USA
[2]Fudan University, China
[3]Purdue University, USA
[4]Rochester Institute of Technology, USA
cuiyiming@ufl.edu, yanliqi@westlake.edu.cn, cao270@purdue.edu, dongfang.liu@rit.edu

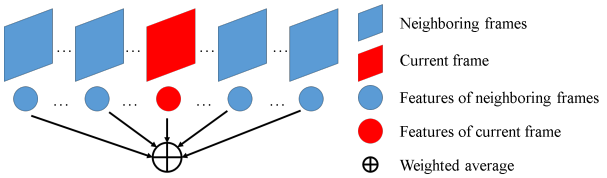
Abstract

Video objection detection is a challenging task because isolated video frames may encounter appearance deterioration, which introduces great confusion for detection. One of the popular solutions is to exploit the temporal information and enhance per-frame representation through aggregating features from neighboring frames. Despite achieving improvements in detection, existing methods focus on the selection of higher-level video frames for aggregation rather than modeling lower-level temporal relations to increase the feature representation. To address this limitation, we propose a novel solution named TF-Blender, which includes three modules: 1) Temporal relation models the relations between the current frame and its neighboring frames to preserve spatial information. 2). Feature adjustment enriches the representation of every neighboring feature map; 3) Feature blender combines outputs from the first two modules and produces stronger features for the later detection tasks. For its simplicity, TF-Blender can be effortlessly plugged into any detection network to improve detection behavior. Extensive evaluations on ImageNet VID and YouTube-VIS benchmarks indicate the performance guarantees of using TF-Blender on recent state-of-the-art methods. Code is available at https://github.com/goodproj13/TF-Blender.

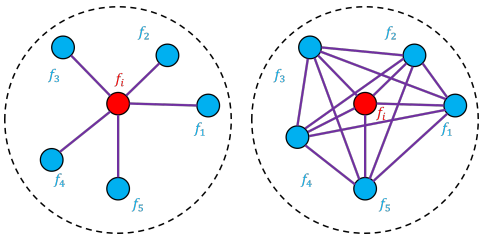
1. Introduction

With the progress of learning-based computer vision, recent research efforts have been extended from image tasks to the more challenging video domains. Video tasks, such as object detection [11], video instance segmentation [40], and multi-object tracking and segmentation [33], hold valuable potentials for real-world applications [24, 33, 25, 26] (i.e., autonomous driving or video surveillance).

A primary challenge of video object detection is to tackle the feature degradation on video frames caused by camera jitter or fast motion. Under the circumstance, detection algorithms for still images are ill-posed for video tasks. Nonetheless, the video has rich temporal information, on which the same object may appear in multiple frames for a certain time span. The value of such temporal information is explored in prior studies using the post-processing paradigm [16, 19, 19, 21]. These methods firstly perform still-image detection on single frames and then assemble the detection results across temporal dimensions using a disjoint post-processing step (i.e., motion estimation and object tracking). None of the above methods, therefore, oper-



(a) Visualization of feature aggregation process



(b) Current aggregation methods (c) Our aggregation methods

Figure 1. Comparison of feature aggregation methods. (a) Features from the neighboring frames are weighted equally during aggregation. (b) The current aggregation methods only reason the relations between the current frame and neighboring frames. (c) Our proposed method computes every pair of frames in the neighborhood in the aggregation process.

*Equal contributions.
†Corresponding author.

ate in an end-to-end fashion. Moreover, if detection on single frames produces weak predictions, the assembling approach cannot improve the detection results.

Alternatively, there have been several attempts to boost the performance of video detection using feature aggregation. [25, 33, 43] leverage optical flow to model the feature movement across frames and propagate temporal features to increase the feature representation for detection. With stronger features, the detection results are significantly improved. However, such temporal features are exploited by an intuitive lumping operation, which is oversimplified.

In terms of how to organize features in aggregation, we recognize two important predecessors, FGFA [42] and SELSA [36]. Compared to the lumping solution [25, 33, 43], both methods use similarity scores to select more helpful features for aggregation. The aggregated feature is organized by an adaptive weight at every spatial location for their representations (as shown in Figure 1(a)). Albeit being superior over the prior efforts, FGFA [42] and SELSA [36] encounter several obstacles from achieving optimal performance: 1) They focus on modeling the global relation for every neighboring frame while ignoring the preservation of the local spatial information for aggregation; 2) They primarily consider the global feature relations to the current frames, while having no constraint in feature learning among the neighboring frames (see Figure 1(b)); 3) They take a fixed number of neighboring frames for the feature aggregation, which is heuristic than general.

In this work, we attempt to take a deeper look at video object detection and improve the performance guarantees by organizing temporal information in a more rigorous principle. Inspired by [42, 36, 7], we propose TF-Blender to organically model features consistently and correspondingly in two ranges. Specifically, we reinforce local similarity in feature space on sequential video frames to depict the continuous and coherence of visual patterns, while identifying semantic correspondence across frames, which makes the temporal representations robust to appearance variations, shape deformations, and local occlusions. In this design, TF-Blender is able to generalize feature aggregation by encouraging the video representation and capturing helpful visual content to improve detection performance. Concretely, we are able to achieve the following contributions:

- We propose a framework called TF-Blender, which depicts the temporal feature relations and blends valuable neighboring features to increase the temporal-spatial feature representation across frames.

- In TF-Blender, we devise a temporal relation module to manage temporal information and a feature adjustment module to add constraints in feature learning to preserve spatial information during feature aggregation. We, therefore, organize the feature learning between every pair of frames and aggregate features in the whole neighborhood (see Figure 1(c))

- Our method is general and flexible, which can be crafted on any detection network. With our novel feature enhancement strategy, we can obtain an absolute gain of more than 0.7% in mAP on the ImageNet VID benchmark and 1.5% in mAP on YouTube-VIS benchmark for recent state-of-the-arts methods.

2. Related Works

2.1. Video Object Detection

Video object detection. Different from image object detection, video object detection faces challenging cases (i.e., motion blur, occlusion, and defocus) which rarely occur in images [8, 15, 44]. To handle the challenges in video domains, several works [20, 19, 16] use post-processing techniques on top of still image detectors. For instance, Seq-NMS [16] links bounding boxes across frames with IoU threshold and re-rank the linked bounding boxes; TCN [20] introduces tubelet modules and applies a temporal convolutional network to embed temporal information to improve the detection across frames; T-CNN [19] applies image object detectors to generate results and then uses optical flow to associate the detected results. Although achieving improvements, none of them are trained end-to-end and their performances are still sub-optimal.

Another focus of the recent works [43, 42, 36, 12, 7, 41, 38] is to aggregate temporal features to improve the feature representation for detection. These methods can be divided into three categories: local aggregation, global aggregation, and combination aggregation. Local aggregation methods [42, 34, 12, 25, 41, 38, 13, 3] usually focus on propagating features in a short range on video sequences. Among them, FGFA [42] and MANet [34] are representatives which use optical flow [18, 14] to calibrate and aggregate features across local frames. On the contrary, global aggregation methods [36, 32, 10] rely on long-range semantic information. One seminal work is from SELSA [36], who computes the semantic similarity between the current frame and its neighbours across the whole video in order to perform temporal feature aggregation. Different from the methods which exploit features locally or globally, MEGA [7] introduces a memory module to use both local and global features to enhance the visual representation of the current frame. The aggregation methods achieve further performance gain over the post-processing methods, but they generally focus on higher-level video frame selection instead of exploring lower-level temporal features exploitation.

Video instance segmentation. Similar to video object detection, MaskTrack R-CNN [40] extends instance segmentation [4, 5] from image domain to video do-

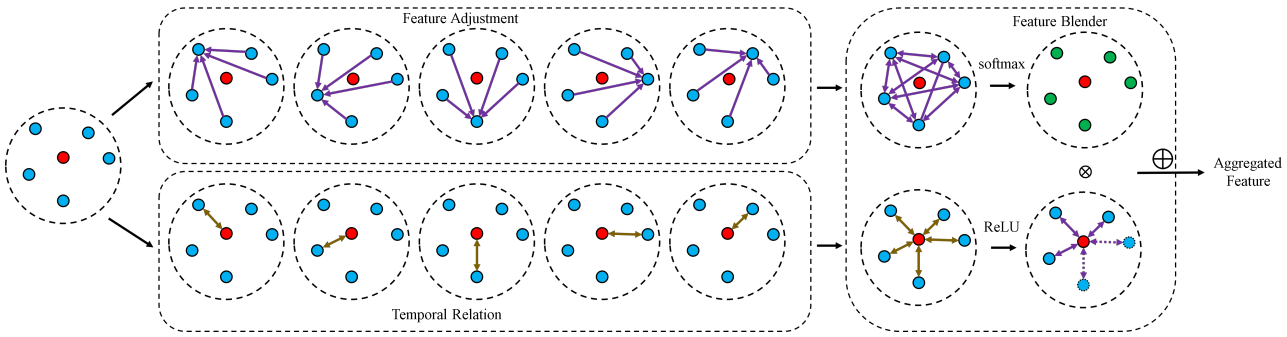


Figure 2. Our TF-Blender framework includes three key modules: a) **Temporal relation module**: Feature relation function $g(f_i, f_j)$ is used as input to learn adaptive weights $\mathcal{W}(f_i, f_j)$ used for feature blender. 2) **Feature adjustment module:** Every neighboring frame feature f_j is aggregated with other neighboring features to generated feature representative $\mathcal{F}(f_i, f_j)$. 3) **Feature blender module:** The results of $\mathcal{W}(f_i, f_j)$ and $\mathcal{F}(f_i, f_j)$ are combined to aggregate the feature of the current frame with dynamic number of neighboring frames.

mains which requires segmenting and tracking instances across frames. However, most of the current methods like MaskProp [2], EnsembleVIS [30] focus on how to track instances across frames rather than how to generate high-quality features for detection, segmentation, and tracking. In this work, we, therefore, propose a more principled solution, which effectively transforms and exploits valuable temporal features for the video object detection task.

2.2. Relation Learning

Relation learning is widely used for different tasks (i.e., point cloud analysis [29, 9] and image understanding [28, 39]) to describe the relationship between the current feature and its neighbors. RS-CNN [29] extends regular grid CNN to capture local point cloud features using geometric topology constraints among points. Similarly, PointConv [37] models the feature relation by computing both the local coordinates and point cloud density. Both methods capture local features in geometric space. On the contrary, DGCNN [35] defines EdgeConv which captures local point relation in high-dimensional feature space and updates the neighborhood for the kernel dynamically at each layer.

Similarly, some recent works attempt to leverage relation learning for object detection. Inspired by [17] which proposes an object relation module for still image object detection, RDN [12] introduces a relation distillation network to aggregate features based on object relation to improving the features for video object detection. MEGA [7] extends the relation learning from RDN and proposes a memory-enhanced global-local aggregation network, which organically manages long-range (global) features and short-range (local) features for aggregation in order to increase the feature representation of current time for detection. However, the focuses of the above methods [17, 12, 7] are the selection of higher-level video frames for aggregation rather than modeling lower-level temporal relation to increasing the feature representation.

Different from these methods, we propose a more general approach for relation learning in feature aggregation. Our TF-Blender can robustly depict the salient correspondences between the feature of the current frame and neighboring frames and exploit only valuable features for a stronger detection.

3. TF-Blender

3.1. Preliminary and Overall Pipeline

The conventional feature aggregation methods [42, 36, 25, 34] generally work in a constrained fashion. Given a set of neighboring frames $\mathbf{F}_j$ of the current frame $\mathbf{F}_i, \forall \mathbf{F}_j \in \mathcal{N}(\mathbf{F}_i)$, their corresponding features f_j are weighted equally based on the feature similarity to $\mathbf{F}_i$ in order to aggregate the temporal feature Δf_i:

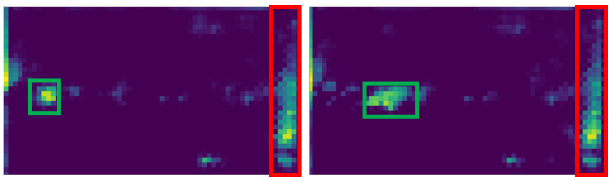
$$\Delta f_i = \sum_{\mathbf{F}_j \in \mathcal{N}(\mathbf{F}_i)} (w_{ij} \times f_j). \tag{1}$$

The principal problem of feature aggregation, therefore, is to calculate weights w_{ij} and select representative neighboring feature f_j. Different from the above simple paradigm, we exploit the temporal features from a general perspective. To achieve this goal, our TF-Blender crafts on three novel architectural modules, temporal relation module, feature adjustment module, and feature blender module, to boot the detection performances (see Figure 2).

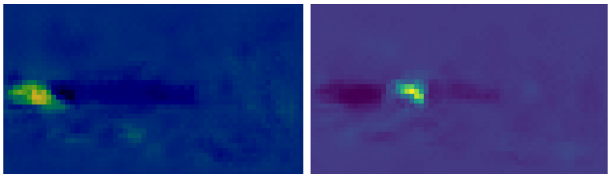
3.2. Temporal Relation

Our temporal relation models the correspondences between the keyframe and its neighbors. To achieve this goal, existing methods use $\mathbb{W}(f_i, f_j)$ to compute a global weight on every pixel in the feature map. This approach ignores local spatial information of the feature map during the process of aggregation, which causes the issue of severe outliers in the feature map. As shown in Figure 3(a), two neigh-

(a) Input frames



(b) Feature maps of input frames



(c) Results of temporal relation

Figure 3. An example of the problem of feature aggregation with global weights: a) shows two neighboring frames where the moving car (the green rectangles) is smaller than the traffic cone (the red rectangles). b) visualizes the feature maps of the two frames where the traffic cone also has a high response besides the car. With global weights, the high response feature of the traffic cone (the red rectangles) cannot be suppressed unless the global weights have very small values. c) shows the results of our proposed temporal relation module which assigns every pixel in the feature map with an adaptive weight and can suppress the irrelevant features.

boring frames have a car with a fast speed and a still traffic cone marked with green and red rectangles respectively. The feature maps of the input frames are visualized as Figure 3(b) and the features of the traffic cone are outliers for the car detection. For global weights, if the weights between the paired frames are none-zero, irrelevant features cannot be removed during aggregation (see Figure 3(b)). This problem occurs frequently when dealing with occlusions or small-scale objects.

To address this issue, our temporal relation module generates adaptive weights $\mathcal{W}(f_i, f_j)$ for every pixel on the feature map in replace of the global weights $\mathbb{W}(f_i, f_j)$. We model $\mathcal{W}(f_i, f_j)$ as a tensor with the same size as the feature representatives for aggregation. For every neighboring frame $\mathbf{F}_j$ of the current frame $\mathbf{F}_i$, we use temporal relation module to calculate adaptive weights $\mathcal{W}(f_i, f_j)$ (see Figure 2). The process is formulated as:

$$\mathcal{W}(f_i, f_j) = \mathcal{M}(g(f_i, f_j)), \tag{2}$$

where g is a feature relation function to describe the temporal relation between f_i and f_j and $\mathcal{M}$ is a masking function

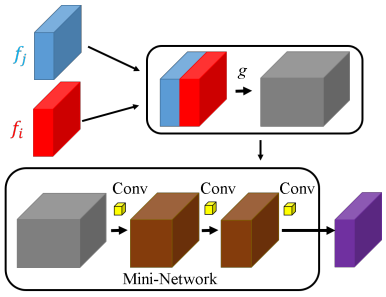


Figure 4. Visualization of temporal relation module. The input feature of f_i and f_j are visualized as blue and red cuboids respectively. Feature relation function g models the temporal relation between f_i and f_j (the gray cuboids). In the mini-network, convolution layers (the yellow cubes) are applied to generate the final results (the purple cuboids). The results of the mid-layers are visualized as brown cuboids.

to calculate the adaptive weight based on g. As shown in Figure 3(c), our temporal relation can enhance the feature representations from the region of interest and suppress the irrelevant features.

More concretely, we compute $\mathcal{M}$ in Eq. 2 using a mini-network (see Figure 4). Compared with the CoefNet in LMP [44], our feature adjustment module is built on a lighter architecture, which makes our TF-Blender computationally efficient. The input of the module is f_i and f_j, marked as red and blue cuboids respectively. Feature relation function g describes the relation between f_i and f_j and generates the input (the gray cuboid) of the mini-network $\mathcal{M}$. Afterward, we apply three convolution layers (the yellow cubes) to generate the final adaptive weights $\mathcal{W}(f_i, f_j)$ (the purple cuboid). The selection of the feature relation function g will be discussed in 4.1.

3.3. Feature Adjustment

Our feature adjustment module aims to represent the feature consistency and salience of the neighboring frames for feature aggregation. A simple solution [42, 36, 7] is to directly use feature f_j from frame $\mathbf{F}_j$ as the follow:

$$\mathcal{F}(f_i, f_j) = f_{j \to i}. \tag{3}$$

However, f_j cannot be guaranteed to be valuable for aggregation as there is no constraints between these neighboring features. Therefore, we aggregate every neighboring frame feature f_j before aggregating the current frame feature f_i. We get feature representative $\mathcal{F}(f_i, f_j)$ by aggregating f_j with the other neighboring features $f_m, \forall \mathbf{F}_m \in \mathcal{N}(\mathbf{F}_i), \mathbf{F}_m \neq \mathbf{F}_j$ (see Figure 2). During feature adjustment, we use the temporal relation module to generate adaptive weights for neighbouring feature aggregation and the

process can be expressed as:

$$\mathcal{F}\left(f_i, f_j\right) = \sum_{\substack{\mathbf{F}_m \in \mathcal{N}(\mathbf{F}_i) \\ \mathbf{F}_m \neq \mathbf{F}_j}} \mathcal{W}\left(f_j, f_m\right) \otimes f_j \tag{4}$$

where $\otimes$ is element-wise multiplication, f_m is the feature of the neighboring frame except itself, and $\mathcal{W}\left(f_j, f_m\right)$ is Eq 2, which can be expressed here as:

$$\begin{aligned} \mathcal{W}\left(f_j, f_m\right) &= \mathcal{M}\left(g\left(f_j, f_m\right)\right) \\ \forall \mathbf{F}_m &\in \mathcal{N}\left(\mathbf{F}_i\right), \mathbf{F}_m \neq \mathbf{F}_j \end{aligned} \tag{5}$$

3.4. Feature Blender

In our feature blender module, we first enhance the results of the temporal relation module with the non-linear function ReLU so that the contrast between the area of interests and background can be captured (see the blender module in Figure 2). We formulate this process as:

$$\hat{\mathcal{W}}\left(f_i, f_j\right) = \texttt{ReLU}\left(\mathcal{W}\left(f_i, f_j\right)\right). \tag{6}$$

Meanwhile, we normalize the results of the feature adjustment module with the softmax function over all the channels to improve the generalization of our model. On the top of the feature blender module in Figure 2, blue dots are normalized to green dots by the softmax function with the guidance of purple double arrows. The process can be expressed as:

$$\hat{\mathcal{F}}\left(f_i, f_j\right) = \texttt{softmax}\left(\mathcal{F}\left(f_i, f_j\right)\right). \tag{7}$$

In our feature blender module, we force $\hat{\mathcal{W}}\left(f_i, f_j\right)$ to be 0 if the adjusted neighboring feature is very similar to the feature of the current frame, shown as dashed purple double arrows in the feature blender module part of Figure 2. We use the cosine distance to describe the similarity between $\hat{\mathcal{F}}\left(f_i, f_j\right)$ and f_i. If the cosine distance is bigger than δ, $\hat{\mathcal{W}}\left(f_i, f_j\right)$ is force to be 0. We define this process as:

$$\hat{\mathcal{W}}\left(f_i, f_j\right) = 0, \quad \textbf{if} \quad \frac{\hat{\mathcal{F}}\left(f_i, f_j\right) \cdot f_i}{|\hat{\mathcal{F}}\left(f_i, f_j\right)||f_i|} > \delta. \tag{8}$$

We have this design because most of the current feature aggregation-based methods [7, 36, 12, 42] have a fixed number of neighboring frames in aggregation. However, for neighboring frames which include issues of severe motion blur or defocus, aggregating them are irrelevant and redundant, which may cause unwanted ambiguity.

Finally, we use element-wise multiplication to combine the results of from Eq. 7 and Eq. 8 to perform the feature aggregation:

$$\Delta f_i = \sum_{\mathbf{F}_j \in \mathcal{N}(\mathbf{F}_i)} \left(\hat{\mathcal{W}}\left(f_i, f_j\right) \otimes \hat{\mathcal{F}}\left(f_i, f_j\right)\right) \tag{9}$$

Methods	mAP(%)	Runtime(FPS)
FGFA[42]	77.8	7.3
SELSA[36]	81.5	10.6
RDN[12]	81.7	-
MEGA[7]	82.9	5.3
FGFA(Ours)	$79.3_{\uparrow 1.5}$	6.9
SELSA(Ours)	$82.5_{\uparrow 1.0}$	10.1
RDN(Ours)	$82.4_{\uparrow 0.7}$	-
MEGA(Ours)	$83.8_{\uparrow 0.9}$	4.9

Table 1. Performance comparison with the recent state-of-the-art video object detection models on ImageNet VID validation set. The backbone is ResNet-101 and runtime is tested on a single RTX 2080Ti GPU.

4. Experiments

4.1. Implementation Details

Evaluation metrics. Following [43, 42], we report all results on using the mean average precision (mAP).

Video object detection setup. We evaluate our methods with MEGA [7], SELSA [36],FGFA [42], and RDN [12], the three state-of-the-art systems. We perform our training and evaluation on the ImageNet VID benchmark [31], which contains 3,862 videos for training and 555 videos for validation. Following the widely used protocols in [42, 7, 36], we train our model on a combination of ImageNet VID and DET datasets. We implement our method mainly based on the source code of the original method. The whole network is trained on 8 RTX 2080Ti GPUs with SGD. During the training and inference process, each GPU holds on one set of images or frames. During the training process, the encoder parameters are frozen and an NMS of 0.5 IoU is adopted to suppress detection redundancy.

Video instance segmentation setup. We also evaluate our proposed method with state-of-the-art MaskTrack R-CNN [22] and SipMask [6]. We perform our training and evaluation on the YouTube-VIS benchmark [40], where there are 3,471 videos for training and 507 videos for validation. During the training process, we use weights pre-trained on MS-COCO [23] and use 8 RTX 6000 GPUs with SGD. In both training and evaluation, the original frame sizes are resized to 640×360.

Parameters. For mini-network $\mathcal{M}$ in Eq. (2), a three-layer CNNs is introduced to adapt the channels for feature aggregation. Feature relation function g is defined as a concatenated tensor of f_i, f_j, $f_i - f_j$, $f_j - f_i$ and the δ in Eq. (8) is set to 0.7.

4.2. Main Results

Results on ImageNet VID benchmarks. We compare state-of-the-art systems crafted on our method with their original implementations. For a fair comparison, we used

Figure 5. Quantitative examples of comparison between methods without and with our TF-Blender integrated on ImageNet VID and YouTube-VIS benchmarks.

the codes provided by the original papers and re-implement them with our proposed method. The results are demonstrated in Table 1. Based on the results, our proposed methods substantially improve the performance of every compared method listed in the table with the same backbone.

For head-to-head comparisons, all the methods with the same backbone can leverage our proposed methods to improve their performances on detection results around 0.7%-1.5% on accuracy. Among them, FGFA with our proposed method has the highest improvement compared with other methods. Among them, local aggregation and global aggregation methods like FGFA [42] and SELSA [36] can have a better improvement with our proposed methods compared with combination aggregation methods like RDN [12] and MEGA [7]. We argue that the limited performance gains come from the combination aggregation methods, which consider both local and global features and make detection more robust to issues like motion blur in videos.

Figure 5 shows some examples of detection results with our methods integrated. Based on the examples, we can see that our proposed method can help solve the problem of weak detection with rare pose and part occlusion situations.

Experiments on YouTube-VIS benchmark. We also evaluate our proposed method on YouTube-VIS dataset [40] and report our results on the validation as [40, 6, 1]. Most of the current video instance segmentation methods focus on how to generate high-quality masks and link the same objects across frames with features extracted by the backbones like ResNet while only a few of them pay attention to improve the features for mask generation and object tracking. We add our proposed methods to these video instance segmentation methods to evaluate the effectiveness of our TF-Blender on issues like motion blur and defocus in videos. The results with ResNet-50 as backbones are shown in Table 2. From Table 2, our proposed methods achieve competitive results under all evaluation metrics. With our proposed methods, MaskTrack R-CNN and SipMask can be improved by more than 1.6% on the AP metric. The bottom part of Figure 5 shows an example of detection and segmentation results with our integrated.

4.3. Ablation Study

We carry out extensive ablation studies to discover the optimal settings related to different settings of our system using FGFA [42].

Analysis of contributing components. We first conduct experiments on the effect of every component in our proposed method and the results are shown in Table 3. The baseline model a is the original FGFA. Every component of our proposed method (temporal relation, feature adjustment, and feature blender) contributes towards improving the overall performance in detection accuracy. By introducing the temporal relation module, the performance of model b can be improved by 0.7%. Model c adds our feature adjustment module to the baseline and gets an improvement of 0.3% compared with the baseline model a. We add our fea-

Methods	Category	AP	AP_{50}	AP_{75}	AR_1	AR_{10}	FPS
Stem-Seg [1]	One-stage	30.6	50.7	33.5	31.6	37.1	12.1
Stem-Seg(Ours)		31.3	51.5	34.1	32.1	37.9	11.3
SipMask [6]		33.7	54.1	35.8	35.4	40.1	28.0
SipMask(Ours)		35.1	55.5	36.9	36.1	41.3	26.6
SG-Net [27]		34.8	56.1	36.8	35.8	40.8	22.9
SG-Net(Ours)		35.7	57.1	37.6	36.6	42.0	21.3
MaskTrack R-CNN [22]	Two-stage	30.3	51.1	32.6	31.0	35.5	10.0
MaskTrack R-CNN(Ours)		31.4	52.3	33.5	31.9	36.5	9.4

Table 2. Performance comparison with the recent state-of-the-art video instance segmentation models on YouTube-VIS validation set. The backbone is ResNet-50-FPN and the models are pretrained on MS-COCO. The runtime is tested on a single RTX TITAN GPU.

Method	TR	FA	FB	mAP(%)
a				77.8
b	✓			78.5
c		✓		78.1
d			✓	78.3
e	✓	✓		78.6
f	✓		✓	78.8
g		✓	✓	78.5
h	✓	✓	✓	79.3

Table 3. Impact of integrating every functional module into the baseline to the accuracy. TR, FA, and FB stand for temporal relation module, feature adjustment module, and feature blender module respectively.

g	mAP(%)
f_i, f_j	78.3
$f_i - f_j$	78.6
$f_i + f_j$	77.9
$f_i, f_j, f_i + f_j$	78.1
$f_i - f_j, f_i + f_j$	78.5
$f_i, f_j, f_i - f_j$	78.9
$f_i, f_j, f_i - f_j, f_j - f_i$	79.3

Table 4. Results of different designs on feature relation function g.

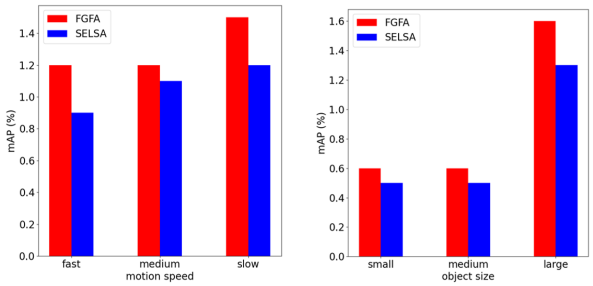
ture blender module to model a to generate dynamic numbers of neighboring frames for feature aggregation and get model d, which is 0.5% better than the original model on mAP metric. Model e, f, and g come from the combination of models a, b and c. As can be shown in Table 3, by combining every two of our proposed methods, the video object detection performance can be further improved. Compared with the baseline model a, our full model h can obtain an absolute gain of 1.5% in accuracy of video object detection.

Analysis of temporal relation. We conduct ablation studies on the choice of g in Eq. (2). During these experiments, all the other experimental settings are kept the same. We first try different combinations of f_i and f_j for g on FGFA [42] as Table 4. A naive idea is to use just f_i and f_j as input and there is 0.5% improvement on FGFA. We think that the performance is limited because only individual frame features are taken into account which is not enough to describe the relationship between the f_i and f_j. Thus, we introduce the difference between f_i and f_j to g and get an improvement of 0.8% for FGFA. We then use the summation of f_i and f_j as g to generate $\mathcal{W}(f_i, f_j)$ but there is only 0.1% improvement. We also make a combination between $f_i + f_j$ with the other choices mentioned above (like f_i, f_j, and $f_i - f_j$), but the results of the combination are worse than those of the original. We think the reason why $f_i + f_j$ is not suitable to describe the relations between f_i and f_j is $f_i + f_j$ works like an average filter which mixes the pixels with higher responses and those with lower responses in the feature map. Besides the experiments mentioned above, we also try $f_i, f_j, f_i - f_j$ and get an improvement of 1.1%. Finally, we choose $f_i, f_j, f_i - f_j, f_j - f_i$ as our feature relation function g, which has the highest detection accuracy. Since f_i and f_j denote the current and adjacent features respectively. Frame $\mathbf{F}_j$ could be a frame before or after the current frame $\mathbf{F}_i$. Thus, it is imperative to calculate both $f_i - f_j$ and $f_j - f_i$, as they model the different temporal correspondence and consistency.

Experiments on $\mathcal{M}$. We conduct experiments on the design of $\mathcal{M}$ for the temporal relation module, especially on the number of layers of $\mathcal{M}$ for the mini-network. Model a is the simplest design where there is only one convolution layer with kernel size 1×1. By keeping the kernel size fixed and adding one more convolution layer, model b can increase the mAP by 0.2%. When there are three convolution layers with kernel size 1×1, the detection accuracy can obtain 79.2% as model c. However, when adding more convolution layers, as in model d, the detection accuracy begins to decrease. We argue that the increasing number of convolution layers introduces arduous parameters in the mini-network which cause overfitting. In model e, we change the kernel size from 1×1 to 3×3 and get an improvement of detection accuracy by 0.1%.

model	# of layers	mAP(%)
a	1	78.8
b	2	79.0
c	3	79.2
d	4	79.1
e	3	79.3

Table 5. Impact of the number of layers for $\mathcal{M}$.



(a) Motion speed (b) Object sizes

Figure 6. Improvement of performance with different motion speeds and object sizes.

Analysis of object sizes and motion speeds. We also investigate the effect of our TF-Blender on the object sizes and motion speeds of the objects. We use the same definition as MS-COCO [23] and FGFA [42] for object sizes and motion speeds respectively. We use mAP as evaluation metrics and visualize the improvement of performance on objects with different sizes and motion speeds as Figure 6. We notice that our method has different improvements on objects with various motion speeds. As shown in Figure 6 (a), there is a higher improvement for objects with slow motion speeds compared with those with fast and medium speeds. We think that there may be two reasons. One is that even though our proposed method can help improve the detection accuracy for objects with fast motion speeds, it's still a challenge to have accurate enough detection results for all the objects with fast-motion speed. Another reason is that objects with slow-motion account for 37.9% in ImageNet VID benchmark while those with medium and fast motion speeds are 35.9% and 26.2% respectively.

Another critical observation from our experiment that our method can offer the highest improvement for detection on large objects, as shown in Figure 6(b). This resonates with the assumption of our proposed method: since large objects have larger feature map sizes, the corresponding pixel can benefit more from an individual weight for fine-grained feature encoding. For small objects, since their feature maps are small, the weights for aggregation have less contribution to feature representation improvement.

Speed-accuracy tradeoff. The computational loads for convectional methods (i.e., FGFA [47] and SELSA [41]) stem from two major sources: 1. feature extraction (encoding) network $\mathcal{N}_{ex}$; 2. task network $\mathcal{N}_{tk}$. Thus, the runtime complexity for the above methods is:

$$\mathcal{O}(\mathcal{N}_{ex}) + \mathcal{O}(\mathcal{N}_{tk}) \tag{10}$$

While the proposed TF-Blender approach is adopted, the computational cost can be defined as:

$$\mathcal{O}(\mathcal{N}_{ex}) + i \cdot \mathcal{O}(\mathcal{N}_{tf}) + \mathcal{O}(\mathcal{N}_{tk}) \tag{11}$$

where $\mathcal{N}_{tf}$ is the cost for the TF-Blender module and i is the number of aggregated frames. Typically, $\mathcal{O}(\mathcal{N}_{tk}) \ll \mathcal{O}(\mathcal{N}_{ex})$ and $\mathcal{O}(\mathcal{N}_{tf}) \ll \mathcal{O}(\mathcal{N}_{ex})$. Thus, the cost ratio r can be expressed as:

$$r = 1 + \frac{i \cdot \mathcal{O}(\mathcal{N}_{tf})}{\mathcal{O}(\mathcal{N}_{ex}) + \mathcal{O}(\mathcal{N}_{tk})} \tag{12}$$

This increasing computational cost is affordable because the impact of $i \cdot \mathcal{O}(\mathcal{N}_{tf})$ is negligible.

We visualize the speed-accuracy tradeoff of FGFA [47] as an example (cf. Figure 7). With the increasing number of input frames, FGFA with TF-Blender achieves significant improvement in accuracy while the runtime increase keeps in an affordable range.

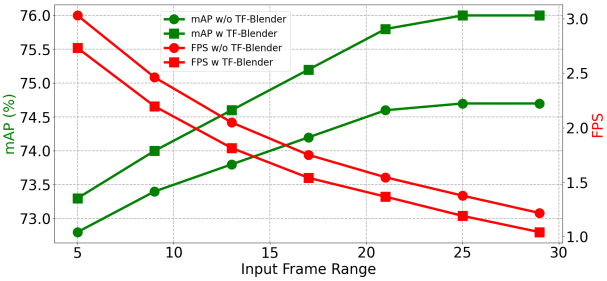


Figure 7. Demonstration of a speed-accuracy tradeoff with and without TF-Blender on FGFA with ResNet-50.

5. Conclusion

In this paper, we discuss the problems of video object detection and introduce a framework named TF-Blender which contains temporal relation, feature adjustment, and feature blender modules to solve the problem of feature degrading in the video frames. Our method is flexible and general, which can be adopted by any learning-based detection network to achieve improved performance. Extensive experiments demonstrate that, with the integration of our proposed method, the current state-of-the-art methods can improve video object detection accuracy on ImageNet VID and YouTube-VIS benchmarks by a large margin. We believe that our TF-Blender can be a valuable addition to the existing methods for temporal feature aggregation for video detection and TF-Blender can be extended to other video analysis tasks like video instance segmentation.

References

[1] Ali Athar, Sabarinath Mahadevan, Aljoša Ošep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos, 2020. 6, 7

[2] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9739–9748, 2020. 3

[3] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks, 2018. 2

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 2

[5] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact++: Better real-time instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[6] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. *arXiv preprint arXiv:2007.14772*, 2020. 5, 6, 7

[7] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. 2020. 2, 3, 4, 5, 6

[8] Daniel Cores, Víctor M. Brea, and Manuel Mucientes. Short-term anchor linking and long-term self-guided attention for video object detection. *Image and Vision Computing*, 110:104179, 2021. 2

[9] Yiming Cui, Xin Liu, Hongmin Liu, Jiyong Zhang, Alina Zare, and Bin Fan. Geometric attentional dynamic graph convolutional neural networks for point cloud analysis. *Neurocomputing*, 432:300–310, 2021. 3

[10] H. Deng, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan. Object guided external memory network for video object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6677–6686, 2019. 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[12] Jiajun Deng, Yingwei Pan, Ting Yao, Wengang Zhou, Houqiang Li, and Tao Mei. Relation distillation networks for video object detection, 2019. 2, 3, 5, 6

[13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect, 2018. 2

[14] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks, 2015. 2

[15] Qichuan Geng, Hong Zhang, Na Jiang, Xiaojuan Qi, Liangjun Zhang, and Zhong Zhou. Object-aware feature aggregation for video object detection, 2020. 2

[16] Wei Han, Pooya Khorrami, Tom Le Paine, Prajit Ramachandran, Mohammad Babaeizadeh, Honghui Shi, Jianan Li, Shuicheng Yan, and Thomas S Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016. 1, 2

[17] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection, 2018. 3

[18] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016. 2

[19] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2017. 1, 2

[20] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. 2

[21] Byungjae Lee, Enkhbayar Erdenee, Songguo Jin, Mi Young Nam, Young Giu Jung, and Phill Kyu Rhee. Multi-class multi-object tracking using changing point detection. In *European Conference on Computer Vision*, pages 68–83. Springer, 2016. 1

[22] Chung-Ching Lin, Ying Hung, Rogerio Feris, and Linglin He. Video instance segmentation tracking with a modified vae architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13147–13157, 2020. 5, 7

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 8

[24] Dongfang Liu, Yiming Cui, Zhiwen Cao, and Yingjie Chen. A large-scale simulation dataset: Boost the detection accuracy for special weather conditions. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. 1

[25] Dongfang Liu, Yiming Cui, Yingjie Chen, Jiyong Zhang, and Bin Fan. Video object detection for autonomous driving: Motion-aid feature calibration. *Neurocomputing*, 409:1–11, 2020. 1, 2, 3

[26] Dongfang Liu, Yiming Cui, Xiaolei Guo, Wei Ding, Baijian Yang, and Yingjie Chen. Visual localization for autonomous driving: Mapping the accurate location in the city maze, 2020. 1

[27] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9816–9825, June 2021. 7

[28] Dongfang Liu, Yiming Cui, Liqi Yan, Christos Mousas, Baijian Yang, and Yingjie Chen. Densernet: Weakly supervised visual localization using multi-scale feature aggregation, 2021. 3

[29] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis, 2019. 3

[30] Jonathon Luiten, Philip Torr, and Bastian Leibe. Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 3

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. 5

[32] M. Shvets, W. Liu, and A. Berg. Leveraging long-range temporal relationships between proposals for video object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9755–9763, 2019. 2

[33] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7942–7951, 2019. 1, 2

[34] Shiyao Wang, Yucong Zhou, Junjie Yan, and Zhidong Deng. Fully motion-aware network for video object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2, 3

[35] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 3

[36] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Sequence level semantics aggregation for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9217–9225, 2019. 2, 3, 4, 5, 6

[37] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds, 2020. 3

[38] Fanyi Xiao and Yong Jae Lee. Video object detection with an aligned spatial-temporal memory, 2018. 2

[39] Liqi Yan, Yiming Cui, Yingjie Chen, and Dongfang Liu. Hierarchical attention fusion for geo-localization, 2021. 3

[40] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5188–5197, 2019. 1, 2, 5, 6

[41] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. Towards high performance video object detection, 2017. 2

[42] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 2, 3, 4, 5, 6, 7, 8

[43] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017. 2, 5

[44] Zhifan Zhu and Zechao Li. Online video object detection via local and mid-range feature propagation. In *Proceedings of the 1st International Workshop on Human-Centric Multimedia Analysis*, HuMA'20, page 73–82, New York, NY, USA, 2020. Association for Computing Machinery. 2, 4