

# Temporal ROI Align for Video Object Recognition

Tao Gong<sup>1,2\*</sup>, Kai Chen<sup>3</sup>, Xinjiang Wang<sup>3</sup>, Qi Chu<sup>1,2†</sup>, Feng Zhu<sup>3</sup>,  
Dahua Lin<sup>4</sup>, Nenghai Yu<sup>1,2</sup>, Huamin Feng<sup>5</sup>

<sup>1</sup> School of Cyberspace Security, University of Science and Technology of China

<sup>2</sup> Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences

<sup>3</sup> SenseTime Research

<sup>4</sup> The Chinese University of Hong Kong

<sup>5</sup> Beijing Electronic Science and Technology Institute

gt950513@mail.ustc.edu.cn, chen kai@sensetime.com, wangxinjiang@sensetime.com, qchu@ustc.edu.cn,  
zhufeng@sensetime.com, dhlin@ie.cuhk.edu.hk, ynh@ustc.edu.cn, oliver.feng@yeah.net

## Abstract

Video object detection is challenging in the presence of appearance deterioration in certain video frames. Therefore, it is a natural choice to aggregate temporal information from other frames of the same video into the current frame. However, ROI Align, as one of the most core procedures of video detectors, still remains extracting features from a single-frame feature map for proposals, making the extracted ROI features lack temporal information from videos. In this work, considering the features of the same object instance are highly similar among frames in a video, a novel Temporal ROI Align operator is proposed to extract features from other frames feature maps for current frame proposals by utilizing feature similarity. The proposed Temporal ROI Align operator can extract temporal information from the entire video for proposals. We integrate it into single-frame video detectors and other state-of-the-art video detectors, and conduct quantitative experiments to demonstrate that the proposed Temporal ROI Align operator can consistently and significantly boost the performance. Besides, the proposed Temporal ROI Align can also be applied into video instance segmentation. Codes are available at <https://github.com/open-mmlab/mmdetection>.

## Introduction

Recently, deep convolutional neural networks have brought great progress in object detection of still images (Ren et al. 2015; Lin et al. 2017; Cai and Vasconcelos 2018; Tian et al. 2019; Duan et al. 2019). Most state-of-the-art single image object detectors usually adopt the region-based detection paradigm (Ren et al. 2015; Cai and Vasconcelos 2018). When directly applying these detectors for video object detection (VID), the detection accuracy suffers from deteriorated object appearances in videos, such as motion blur, video defocus and object occlusions. Despite these challenges, video contains temporal information about the same object instance (e.g., its appearance in different poses, and from different viewpoints). Therefore, the key challenge is how to effectively exploit such temporal information for the same object instance in a video.

As shown in Fig. 1 (a), region-based detectors usually adopt ROI Align (He et al. 2017) to extract ROI features. However, ROI Align only utilizes current frame feature map to extract features for current frame proposals, this leads to the extracted ROI features lacking the temporal information of the same object instance in a video. One simple and obvious way to utilize the temporal information is taking other frames feature maps to perform ROI Align for current frame proposals. But the precise location of the current frame proposals in other frame feature maps is unknown, making the simple method infeasible.

In fact, there are already many previous works (Zhu et al. 2017a; Wu et al. 2019; Xiao and Jae Lee 2018; Liu and Zhu 2018; Chen et al. 2018; Deng et al. 2019a) attempting to exploit temporal information in videos. Some works try to utilize image level information from nearby frames to help detection. For example, FGFA (Zhu et al. 2017a), MANet (Wang et al. 2018a) utilize optical flow (Dosovitskiy et al. 2015), and STSN (Bertasius, Torresani, and Shi 2018) applies deformable convolutions (Dai et al. 2017) for image feature calibration. PS ROI Pooling (Dai et al. 2016) is used to pool features for proposals from the calibrated image feature map in these methods. The extracted PS ROI features contain temporal information from nearby frames. However, these methods can only utilize nearby frames within one second (usually at most 30 frames). Performance will degrade with longer time interval, making it hard to leverage information from frames that are far apart in time. Other works attempt to utilize proposal level information from longer video length. SELSA (Wu et al. 2019) and (Shvets et al. 2019) aggregate the high-level proposal features (fully connection layer features of proposals) with each other in order to make every proposal features in current frame contain the high-level proposal features from other frames. However, the ROI features are still extracted from a single image.

In this paper, a novel operator, named as Temporal ROI Align, is proposed to exploit the temporal information for the same object instance in a video. As shown in Fig. 1 (b), we define a target frame as a frame where final prediction is done at the moment. The target frame is allowed to have multiple support frames, which are used for strengthening the features of the target frame. Considering the features of

\*Work done during an internship at SenseTime.

†Qi Chu is the corresponding author.

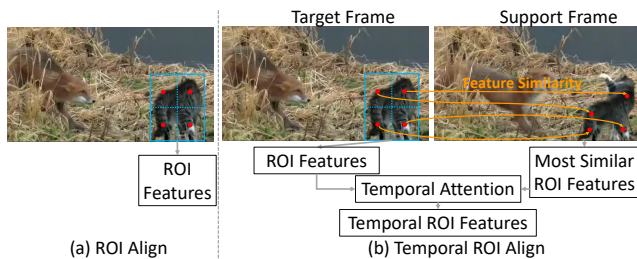


Figure 1: The Illustration of Temporal ROI Align. (a) The conventional ROI Align. The spatial size of ROI features is set to  $2 \times 2$  for convenience. (b) Based on feature similarity, Temporal ROI Align implicitly extracts the most similar ROI features from support frames feature maps for target frame proposals. Then, temporal attention mechanism is conducted to aggregate the ROI features and the most similar ROI features in order to generate final temporal ROI features

the same object instance are highly similar among frames in a video, the proposed operator implicitly extracts the most similar ROI features from support frames feature maps for target frame proposals based on feature similarity. The extracted most similar ROI features contain the temporal information of the same object instance in a video. The key challenge now is how to effectively aggregate these ROI features. One simple and obvious way is to average them. However, it is suboptimal due to the fact that an object instance may be blurry in some frames and clear in other frames. It is obvious that the ROI features of clear object instances should play a more important role than features of blurry object instances during aggregation. Therefore, temporal attention mechanism is conducted to aggregate the ROI features and the most similar ROI features.

The proposed operator also inherits the advantage of SELSA (Wu et al. 2019) and (Shvets et al. 2019) which leverage long-range temporal information of a video. It can also be applied into other video object detectors and other video tasks of computer vision, such as video instance segmentation (VIS).

The contributions of the paper are summarized as follows:

1. A novel Temporal ROI Align operator is proposed to offer an alternative for the conventional ROI Align operator in videos. It can extract temporal information from an entire video with arbitrary length for a proposal.

2. We integrate the proposed Temporal ROI Align into single-frame video detectors and other state-of-the-art video detectors, and demonstrate through quantitative experiments that the proposed Temporal ROI Align operator can consistently and significantly boost the performance.

3. Experiments show that the proposed Temporal ROI Align can also be used in other video tasks of computer vision, such as video instance segmentation (VIS).

## Related Work

### Object Detection In Still Images

Most state-of-the-art object detectors (Girshick et al. 2014; Girshick 2015; Ren et al. 2015; Dai et al. 2016) usually

adopt two stage framework composed of a proposal generator and a region classifier. Fast R-CNN (Girshick 2015) develops a ROI Pooling layer to extract features of proposals as the input of region classifier. Faster R-CNN (Ren et al. 2015) proposes Region Proposal Network (RPN) to combine proposal generation with region classification into one framework. The methods mentioned above usually adopt ROI Align (He et al. 2017) or ROI Pooling to extract features for proposals within a single image. As a consequence, when applying these single image detectors to VID, the features extracted from ROI Align lack the temporal information in videos.

Recently, self-attention mechanisms (Vaswani et al. 2017) have been proven to be beneficial for single image recognition. Object relation module (Hu et al. 2018) is proposed to model the proposal relationship within a single image. (Gu et al. 2018) proposes a method which can extract features for proposals using the features of a whole image rather than only the features covered by proposals. Different from these methods which attempt to interact high-level features among objects (e.g. computer vs mouse in (Hu et al. 2018)), we focus on extracting features from a video for the same object instance.

### Object Detection In Videos

Existing methods usually perform image-level or proposal-level feature aggregation to exploit temporal information from a video for the same object instance.

**Image-level features across frames.** There has been a line of research that boosts the single-frame detector by utilizing image-level features. D&T (Feichtenhofer et al. 2017) builds a dense correlation map between two feature maps of nearby video frames, and exploits instance track ids to learn frame-to-frame motion of bounding boxes. DFF (Zhu et al. 2017b) firstly uses optical flow to propagate and align the features of selected keyframes to nearby non-keyframes in order to reduce redundant calculation and speed up the single-frame detector. FGFA (Zhu et al. 2017a) utilizes optical flow to aggregate the nearby frames feature maps to target frame feature map in order to improve the single-frame detector. Furthermore, (Zhu et al. 2018) designs more advanced image-level feature propagation and keyframe selection mechanisms to achieve a better trade-off between accuracy and speed than FGFA. In order to avoid explicit optical flow computation, STSN (Bertasius, Torresani, and Shi 2018) proposes using deformable convolution to compute the offsets for feature alignment. (Wang et al. 2018b) proposes non-local module in order to completely drop the locality of estimated correspondences. However, the performance of these methods degrades quickly with longer time interval. In contrast to these methods, the proposed Temporal ROI Align doesn't perform image-level feature alignment and aggregation, and can take advantage of longer video length (even the entire video) than these methods.

**Proposal-level features across frames.** Another line of research attempts to aggregate the proposal-level features across frames. MANet (Wang et al. 2018a) firstly utilizes optical flow to propagate current frame proposals to nearby frames, then aggregates the features of proposals from multi-

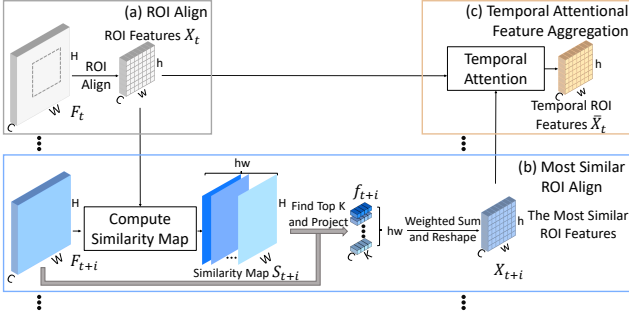


Figure 2: Temporal ROI Align. Firstly, the ROI features  $X_t$  is extracted from *target* frame feature map  $F_t$  for target frame proposal in module (a). Then, Most Similar ROI Align (MS ROI Align) extracts the most similar ROI features  $X_{t+i}$  from *support* frame feature map  $F_{t+i}$  for target frame proposal in module (b). Finally, Temporal Attentional Feature Aggregation (TAFA) performs temporal attention to aggregate  $\{X_{t+i}\}_{i=-T/2}^{T/2}$  in module (c)

frames. RDN (Deng et al. 2019b) introduces the object relation module from still images into videos. It builds two stage relation modules to fully exploit the relationships among proposals. As mentioned above, the performance of these methods degrades quickly with longer time interval. Subsequent works attempt to overcome the shortcomings, i.e. previous works can’t leverage long-range temporal information in videos. SELSA (Wu et al. 2019) firstly generates proposals of multi-frames, and aggregates the high-level proposal features with each other. (Shvets et al. 2019) takes a further step. It develops a loss function which constraints the the feature aggregation conducted among the proposals belong to the same object instance. Although these methods perform high-level proposals’ feature aggregation to utilize the temporal information in videos, they still adopt ROI Align operation to extract the ROI features for proposals, making the extracted ROI features lack temporal information from videos. In contrast to these methods, the proposed Temporal ROI Align can directly extract the temporal information from videos for proposals. Besides, the proposed operator can also be applied into these methods to further boost the performance.

## Temporal ROI Align

The way to extract the features of proposals is crucial in videos. Existing methods usually adopt ROI Align operation to implement it. However, ROI Align is designed to extract the features of proposals from a single image features, making the extracted ROI features lack the temporal information for video tasks.

We propose Temporal ROI Align operator as an alternative to make the features of proposals contain temporal information of videos. Given a set of frames  $\{I_{t+i}\}_{i=-T/2}^{T/2}$  from the same video, a set of feature maps  $\{F_{t+i}\}_{i=-T/2}^{T/2}$  is extracted by a backbone network  $g_{cnn}(\cdot; \theta_{cnn})$ :

$$F_{t+i} = g_{cnn}(I_{t+i}; \theta_{cnn}) \quad (1)$$

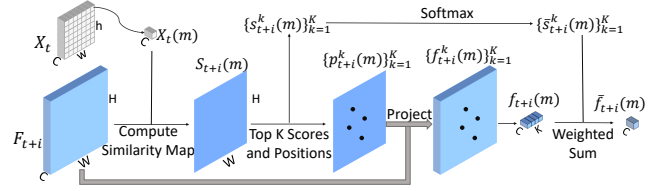


Figure 3: The technical detailed illustration of MS ROI Align in the proposed Temporal ROI Align

where  $t$  denotes the index of target frame and  $t+i$  ( $i \neq 0$ ) denotes the index of support frames.  $T$  is the total number of support frames. As shown in Fig. 2, firstly, the ROI features  $X_t$  is extracted for target frame proposals from *target* frame feature map  $F_t$  by the conventional ROI Align operator in module (a). Then, Most Similar ROI Align (MS ROI Align) focuses on extracting the most similar ROI features  $X_{t+i}$  from *support* frame feature map  $F_{t+i}$  for target frame proposals in module (b). Specifically, similarity maps are calculated between  $F_{t+i}$  and each spatial location of  $X_t$ . For each similarity map, we find the top K similarity scores as the most similar points, and project these points into  $F_{t+i}$ . Based on these points, the most similar features  $f_{t+i}$  can be extracted from  $F_{t+i}$ .  $f_{t+i}$  is weightedly summed by the normalized top K similarity scores to generate the most similar ROI features  $X_{t+i}$ . Finally, Temporal Attentional Feature Aggregation (TAFA) utilizes temporal attention to aggregate  $\{X_{t+i}\}_{i=-T/2}^{T/2}$  in order to generate the final temporal ROI features  $\bar{X}_t$  in module (c). The temporal ROI features  $\bar{X}_t$  contain the object features from target frame and support frames in a video.

## Most Similar ROI Align

MS ROI Align aims at extracting the most similar ROI features from support frames feature maps for target frame proposals. Considering the features of an object instance are highly similar among different frames, we propose to implicitly extract the ROI features from support frames feature maps for target frame proposals based on feature similarity.

As shown in Fig. 3, the input is ROI features  $X_t \in \mathbb{R}^{h \times w \times C}$  and the support frame feature map  $F_{t+i} \in \mathbb{R}^{H \times W \times C}$ .  $h, w$  and  $C$  denote the height, width and channel of  $X_t$  respectively.  $H$  and  $W$  denote the height and width of  $F_{t+i}$  respectively. One spatial location  $X_t(m)$  is used to describe the technical details, where  $m$  denotes the spatial location of  $X_t$  and  $m \in \{(1, 1), (1, 2), \dots, (h, w)\}$ . Other spatial locations of  $X_t$  are performed in the same way.

Firstly both  $X_t(m)$  and  $F_{t+i}$  are L2 normalized along with the channel dimension to generate  $\hat{X}_t(m)$  and  $\hat{F}_{t+i}$ . Then, cosine similarity map  $S_{t+i}(m) \in \mathbb{R}^{H \times W}$  is calculated as follows:

$$S_{t+i}(m) = S(\hat{X}_t(m), \hat{F}_{t+i}) : \mathbb{R}^{1 \times C} \otimes [\mathbb{R}^{(H \times W) \times C}]^T \rightarrow \mathbb{R}^{H \times W} \quad (2)$$

where  $\otimes$  denotes matrix multiplication and  $[\cdot]^T$  denotes matrix transposition.

Then, the top  $K$  similarity scores  $\{s_{t+i}^k(m)\}_{k=1}^K \in \mathbb{R}^K$  and corresponding spatial locations  $\{p_{t+i}^k(m)\}_{k=1}^K \in \mathbb{R}^{K \times 2}$

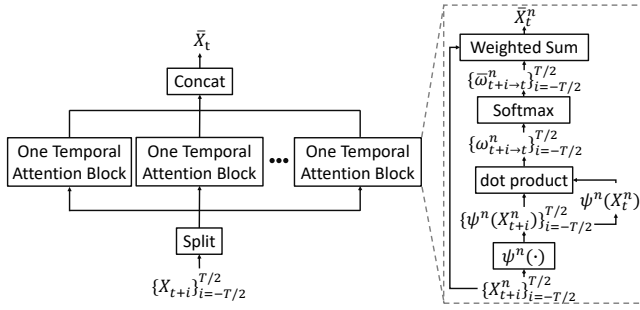


Figure 4: Temporal Attentional Feature Aggregation in the proposed Temporal ROI Align

are found from the similarity map  $S_{t+i}(m) \in \mathbb{R}^{H \times W}$ . The locations  $\{p_{t+i}^k(m)\}_{k=1}^K$  are projected into the support frame feature map  $F_{t+i}$  in order to extract the most similar features  $\{f_{t+i}^k(m)\}_{k=1}^K$  for  $X_t(m)$ .  $f_{t+i}(m)$  is defined as the set of  $\{f_{t+i}^k(m)\}_{k=1}^K$  and  $f_{t+i}(m) \in \mathbb{R}^{K \times C}$ .

Finally, the similarity scores  $\{s_{t+i}^k(m)\}_{k=1}^K$  are used to weight the  $f_{t+i}(m)$ :

$$\bar{s}_{t+i}^k(m) = \frac{\exp(s_{t+i}^k(m))}{\sum_{k=1}^K \exp(s_{t+i}^k(m))} \quad (3)$$

$$\bar{f}_{t+i}(m) = \sum_{k=1}^K \bar{s}_{t+i}^k(m) f_{t+i}^k(m) \quad (4)$$

where  $\bar{s}_{t+i}^k(m)$  is the normalized weight of  $s_{t+i}^k(m)$ .  $\bar{f}_{t+i}(m) \in \mathbb{R}^C$  is the weighted most similar features of  $X_t(m)$ .

Since there are total  $h \times w$  spatial locations of ROI features  $X_t$ , the final most similar ROI features  $X_{t+i} \in \mathbb{R}^{h \times w \times C}$  can be extracted by concatenating all  $\bar{f}_{t+i}(m)$ . The most similar ROI features  $X_{t+i}$  can be regarded as the ROI features extracted from support frame feature map  $F_{t+i}$  for the target frame proposals, since  $X_{t+i}$  is the most similar features with  $X_t$  in each spatial location.

### Temporal Attentional Feature Aggregation

In the MS ROI Align, we have already extracted the ROI features from target frame feature map and support frames feature maps. The remaining key challenge now becomes how to effectively aggregate these ROI features. Since an object instance may be blurry in some frames and clear in other frames, it is natural to learn a group of temporal attention weights to aggregate them. Besides, multi-head attention (Vaswani et al. 2017) allows model to jointly attend information from different representation subspaces at different channels. Therefore, we constitute multi temporal attention blocks to deal with different patterns in the temporal feature aggregation.

As shown in Fig. 4, the input is the set of ROI features  $\{X_{t+i}\}_{i=-T/2}^{T/2}$ . There are  $N$  temporal attention blocks to aggregate  $\{X_{t+i}\}_{i=-T/2}^{T/2}$ . Firstly,  $\{X_{t+i}\}_{i=-T/2}^{T/2}$  are split into  $N$  groups along the channel dimension:

$$X_{t+i}^n = X_{t+i}[:, :, (n-1)\frac{C}{N} : n\frac{C}{N}] \quad (5)$$

where  $X_{t+i}^n \in \mathbb{R}^{h \times w \times \frac{C}{N}}$  and  $n \in \{1, 2, \dots, N\}$ . Each  $\{X_{t+i}^n\}_{i=-T/2}^{T/2}$  is used to generate one attention map:

$$w_{t+i \rightarrow t}^n(m) = \psi^n(X_{t+i}^n)(m) \cdot \psi^n(X_t^n)(m) \quad (6)$$

$$\bar{w}_{t+i \rightarrow t}^n(m) = \frac{\exp(w_{t+i \rightarrow t}^n(m))}{\sum_{i=-T/2}^{T/2} \exp(w_{t+i \rightarrow t}^n(m))} \quad (7)$$

where  $\psi^n(\cdot)$  is a tiny embedding network (network weights are not shared across different temporal attention blocks) and  $\psi^n(X_{t+i}^n) \in \mathbb{R}^{h \times w \times \frac{C}{N}}$ .  $m$  denotes the spatial position of  $\psi^n(X_{t+i}^n)$  and  $\psi^n(X_{t+i}^n)(m) \in \mathbb{R}^{\frac{C}{N}}$ .  $w_{t+i \rightarrow t}^n \in \mathbb{R}^{h \times w}$  and  $w_{t+i \rightarrow t}^n(m)$  is the attention weights from  $t+i$  frame to  $t$  frame of the  $n$ -th group attention map on spatial location  $m$ .  $\bar{w}_{t+i \rightarrow t}^n(m)$  is the normalized attention weights across frames. Based on the normalized  $n$ -th group of temporal attention map  $\{\bar{w}_{t+i \rightarrow t}^n\}_{i=-T/2}^{T/2}$ , the  $n$ -th group of ROI features  $\{X_{t+i}^n\}_{i=-T/2}^{T/2}$  are weightedly summed as follows:

$$\bar{X}_t^n(m) = \sum_{i=-T/2}^{T/2} \bar{w}_{t+i \rightarrow t}^n(m) X_{t+i}^n(m) \quad (8)$$

where  $\bar{X}_t^n \in \mathbb{R}^{h \times w \times \frac{C}{N}}$ . The final temporal ROI features  $\bar{X}_t$  can be obtained by concatenating all  $\bar{X}_t^n$  along the channel dimension and  $\bar{X}_t \in \mathbb{R}^{h \times w \times C}$ . The  $\bar{X}_t$  is the same size as  $X_t$ , but it contains the temporal information of the same object instance in a video.

In some cases like object disappears in some support frames, similar features of MS ROI Align from these supporting frames may be "incorrect". The proposed TAFE can alleviate the issue by the attention weight. The "incorrect" features can be suppressed by the softmax operator if "correct" features (i.e. object features) can be extracted from other support frames. Overall, the MS ROI Align is responsible for extracting the "correct" object features as far as possible. If it fails, the TAFE will minimize the adverse influence of the "incorrect" features by the attention weight. The MS ROI Align with TAFE guarantees that the proposed Temporal ROI Align works on most cases.

Temporal ROI Align is designed as a general operator to extract better features for objects, and it can also work under some special cases. For example, when an object in target frame is partial occlusion, Temporal ROI Align can still extract the object features from support frames for most spatial locations without occlusion. Therefore, the visible parts are dominant and features from these locations can still get enhanced.

## Experiments On VID

### DataSet And Evaluation Metric

Experiments are carried out on the ImageNet VID dataset (Russakovsky et al. 2015) which contains 30 object categories for video object detection. There are total 3862 video snippets in the training set and 555 video snippets in the validation set. The mAP@IoU=0.5 is reported on the validation set.



## Implementation Details

**Backbone Network.** The ResNet-101 (He et al. 2016) (R101) is used as the backbone network  $g_{cnn}(\cdot; \theta_{cnn})$  for ablation studies. ResNeXt-101-64x4d (Xie et al. 2017) (X101) is also used for the final results. The stride of the first conv block in the *conv5* stage of convolutional layers is modified from 2 to 1 in order to enlarge the resolution of feature maps. As such, the effective stride in the stage is changed from 32 pixels to 16 pixels. All the  $3 \times 3$  conv layers in the stage are modified by the dilated convolutions to compensate the receptive fields.

**Region Proposal Network.** RPN is placed on the output of *conv4* to generate proposals. There are a total of 12 anchors with 4 scales  $\{64^2, 128^2, 256^2, 512^2\}$  and 3 aspect ratios  $\{1:2, 1:1, 2:1\}$ . 300 proposals are produced on each image.

**Temporal ROI Align.** We replace the conventional ROI Align operator with the proposed Temporal ROI Align operator. The Temporal ROI Align is applied on the output of *conv5*. The spatial size  $h, w$  of the ROI features are set to 7. For MS ROI Align, there are a total of 49 similarity maps needed to be calculated for each proposal, and the top 4 ( $K = 4$ ) similarity scores and corresponding spatial locations are selected for each similarity map. For TAFA, a total of 4 ( $N = 4$ ) temporal attention blocks are used to aggregate the ROI features and these most similar ROI features, and each  $\psi^n(\cdot)$  is a  $3 \times 3$  convolution layer.

**Detection Network.** Two 1024-d fully connected layers are applied upon the temporal ROI features followed by classification and bounding box regression.

**Training and Testing Details.** The backbone networks are initialized with ImageNet pre-trained weights. A total of 6 epochs of SGD training is performed with a total batch size of 16 on 16 GPUs. The initial learning rate is 0.02 and is divided by 10 at the 4-th and 6-th epoch. The models are trained with a mixture of ImageNet VID and ImageNet DET datasets (Russakovsky et al. 2015) (using 30 VID classes and the overlapping 30 of 200 DET classes). For fair comparison, we use the split provided in FGFA (Zhu et al. 2017a). At most 15 frames are subsampled from each video, and the DET:VID balance is approximately 1:1. For training, one training frame is sampled along with two random frames from the same video. For inference,  $T$  frames (support frames) from the same video are sampled along with the inference frame (target frame). Following STSN [1], if support frames are beyond the video start/end, we copy the first/last frame of the video. NMS with a threshold of 0.5 is adopted to suppress reduplicate detection boxes. In both training and inference, the images are resized to a shorter side of 600 pixels.

## Ablation Study

**Support Frame Sampling Strategies.** The sampling strategy for support frame is crucial for video object detection. Here, we conduct experiments to analyse the effect of the number of support frames  $T$  and sampling frame strides  $S$ . Specifically, by using a sampling frame stride  $S$ , one frame in every  $S$  frames is used for testing rather than consecutive frames.

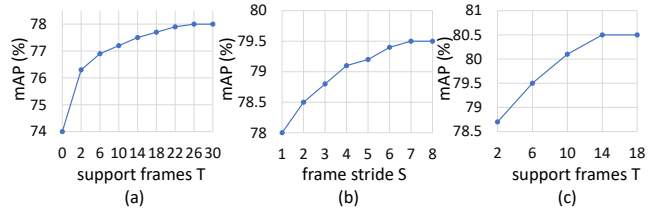


Figure 5: Ablation for different support frame sampling strategies. (a) The effect of different number of support frames  $T$ . The frame stride  $S$  is fixed to 1. (b) The effect of different sampling frame stride  $S$ . The number of support frames  $T$  is fixed to 26. (c) The effect of different number of support frames  $T$ . The support frames  $T$  are uniformly sampled from the entire video

Firstly, we fix the stride  $S$  to 1 and vary the number of  $T$  to analyse the effect of the number of consecutive support frames. As shown in Fig. 5 (a), the single-frame detector baseline, which only uses the conventional ROI Align to extract features from target frame feature map, achieves 74.0 mAP without any support frames. As the number of support frames increases, the performance can be improved consistently and finally stabilized at 78.0 mAP after  $T = 26$ . It means that using more consecutive support frames usually can bring more performance gains.

Then, we fix the number of support frames  $T$  to 26 and vary the number of  $S$  to explore the effect of the fixed sampling stride. As shown in Fig. 5 (b), the performance increases steadily with  $S$  raising and finally stabilizes on 79.5 mAP after  $S = 7$ . It indicates that sampling support frames with a larger stride can further improve the performance since longer-range temporal information can be leveraged, which also coincides the results in (Shvets et al. 2019) and (Wu et al. 2019).

Finally, to further utilize the whole video information, we uniformly sample the support frames from the entire video, i.e. the sampling stride  $S$  is adaptive to the length of the video according to the number of the support frames  $T$ . Fig. 5 (c) shows the performance with different values of  $T$ . We can see that with 2 support frames that are uniformly sampled from the entire video, the performance already achieves 78.7, which outperforms the 26 consecutive support frames by 0.7 point. With 6 support frames that are uniformly sampled from the entire video, the performance can achieve the same results with 26 support frames that are sampled by the fixed sampling frame stride 8. The performance keeps increasing as  $T$  rises and finally stabilizes at 80.5 mAP after  $T = 14$ . These results demonstrate that the proposed Temporal ROI Align can utilize long-range temporal information from videos. The uniform sampling strategy with  $T = 14$  is used as the default setting in the following experiments.

**The Effectiveness of MS ROI Align and TAFA.** MS ROI Align in the proposed Temporal ROI Align is designed to find the location corresponding to the proposal of target frame from the support frame feature maps. Inspired by STSN (Bertasius, Torresani, and Shi 2018) that utilizes deformable convolutions to calculate the offsets used for image feature calibration, we replace the ROI Align with the

Method	mAP(%)
Single-frame Detector	74.0
MD ROI Align + Averaging	68.5
MD ROI Align + TAFA	76.6
MS ROI Align + Averaging	78.5
MS ROI Align + TAFA	<b>80.5</b>

Table 1: The Effectiveness of MS ROI Align and TAFA. "MD ROI Align" denotes that the MS ROI Align is replaced by Modulated Deformable ROI Align. "Averaging" denotes the TAFA is replaced by simply averaging

Method	mAP(%)
Single-frame Detector	74.0
ROI Align + Non-Local	77.9
ROI Align + Non-Local (Top K)	78.4
Temporal ROI Align	<b>80.5</b>

Table 2: Performance comparison with Non-local and Temporal ROI Align

recently proposed Modulated Deformable ROI Align (Zhu et al. 2019) (MD ROI Align) to show the effectiveness of MS ROI Align. The MD ROI Align can calculate the offsets between target frame and support frames in order to extract the ROI features from support frames feature maps for target frame proposals. As shown in Table 1, "MD ROI Align + TAFA" achieves 76.6 mAP, which only exceeds the single-frame baseline by 2.6 points. The reason behind this limited improvement lies in that deformable operation can't learn the effective offsets when the frame stride is large, which is also shown in the results of STSN (Bertasius, Torresani, and Shi 2018) where the performance degrades from 78.9 to 77.9 when the frame stride changes from 1 to 4. Compared to "MS ROI Align + TAFA", the performance of "MD ROI Align + TAFA" drops around 4 points, which demonstrates the superiority of the designed MS ROI Align.

To demonstrate the effectiveness of TAFA for feature aggregation, we also conduct a simple baseline where the ROI features from target frame and support frames are directly averaged. The experiment results are shown in Table 1. We can see that "MS ROI Align+ Averaging" only achieves 78.5 mAP, which is lower than "MS ROI Align+ TAFA" by 2 points. This demonstrates that the temporal attention design in TAFA can learn to put more weights on better ROI features.

**The Effectiveness of Temporal ROI Align.** Since Non-Local (Wang et al. 2018b) can also extract the temporal information from support frames, we also attempt to integrate ROI Align with Non-Local module in order to show the superiority of Temporal ROI Align. As shown in Table 2, the mAP is only 77.9 when applying Non-Local after ROI Align. Non-local inevitably aggregates background features into the ROI features, since the attention weights of Non-Local are computed from all spatial locations in support frames. This leads to the inferior mAP of ROI Align + Non-Local. We also modify the original Non-Local by pick-

# $K$	1	2	3	4	5	6
mAP(%)	79.3	79.8	80.2	<b>80.5</b>	80.4	80.2

Table 3: The effect of different the most similar K locations in MS ROI Align of Temporal ROI Align

# $N$	1	2	4	8	16	32
mAP(%)	79.5	80.1	<b>80.5</b>	<b>80.5</b>	80.4	80.3

Table 4: The effect of different number of the temporal attention blocks  $N$  in TAFA of Temporal ROI Align

ing the Top K attention weights to eliminate the background features as far as possible. However, the result (78.4) is still lower than the proposed Temporal ROI Align (80.5) by 2.1 points. This shows the superior performance of the proposed Temporal ROI Align.

**Experiments on Hyperparameters.** We analyse the effect of two hyperparameters  $K$  and  $N$ , standing for the number of the most similar locations on each similarity map and the number of the temporal attention blocks respectively.

Table 3 shows the effect of different  $K$ . With the  $K$  increasing from 1 to 4, the performance increases from 79.3 mAP to 80.5 mAP. This shows that the extracted most similar ROI features can benefit from more sampling location  $K$ . However, the performance seems to have a downward trend when  $K$  is larger than 4, since using large  $K$  may find the features of unrelated location, especially for small objects. Therefore,  $K$  is chosen to 4 as default setting.

Table 4 shows the effect of using different number of the temporal attention blocks  $N$  in TAFA. With the  $N$  increasing from 1 to 4, the performance increases from 79.5 mAP to 80.5 mAP. This shows that using more temporal attention blocks can improve the accuracy. The performance saturates when  $N$  is bigger than 4. Therefore,  $N$  is chosen to 4 as default setting.

## Comparison With State-of-the-art Methods

The proposed Temporal ROI Align offers an alternative for the conventional ROI Align operator. Therefore, it is easy to be implemented into other video detectors. We reproduce the newly proposed SELSA (Wu et al. 2019) due to its effectiveness, simpleness and also the state-of-the-art performance. As shown in Table 5, SELSA-*ReIm* denotes the reimplementation of SELSA, and it basically achieves the same performance with SELSA (Wu et al. 2019) under ResNet-101 and ResNeXt-101 backbones. When replacing ROI Align by the proposed Temporal ROI Align, the performance further improves by 1.7 points and 1.3 point respectively, which demonstrates the flexibility of the proposed operator.

Table 6 summarizes the performance of the proposed method and other state-of-the-art models on the ImageNet VID validation set. Without bells and whistles, the proposed Temporal ROI Align can achieve 80.5 mAP with ResNet-101 backbone when applied to single-frame detector baseline. When applied to SELSA (Wu et al. 2019), the performance further improves to 82.0 mAP. Furthermore, by using

Method	Backbone	mAP(%)
SELSA (Wu et al. 2019)	R101	80.25
SELSA- <i>ReIm</i>	R101	80.3
SELSA- <i>ReIm</i> + TROI	R101	<b>82.0</b>
SELSA (Wu et al. 2019)	X101	83.11
SELSA- <i>ReIm</i>	X101	83.0
SELSA- <i>ReIm</i> + TROI	X101	<b>84.3</b>

Table 5: Applying the Temporal ROI Align (TROI) into SELSA video detector. Note that the SELSA-*ReIm* denotes the reimplementation of SELSA

Method	Backbone	mAP(%)
FGFA (Zhu et al. 2017a)	R101	78.4
D&T (Feichtenhofer et al. 2017)	R101	80.0
PLSA (Guo et al. 2019)	R101+DCN	80.0
SELSA (Wu et al. 2019)	R101	80.25
Leveraging (Shvets et al. 2019)	R101-FPN	81.0
RDN (Deng et al. 2019b)	R101	81.8
MEGA (Chen et al. 2020)	R101	82.9
<b>Single-frame Detector + TROI</b>	R101	<b>80.5</b>
<b>SELSA (Wu et al. 2019) + TROI</b>	R101	<b>82.0</b>
D&T (Feichtenhofer et al. 2017)	X101	81.6
SELSA (Wu et al. 2019)	X101	83.11
RDN(Deng et al. 2019b)	X101	83.2
Leveraging (Shvets et al. 2019)	X101-FPN	84.1
MEGA (Chen et al. 2020)	X101	84.1
MEGA <sup>△</sup> (Chen et al. 2020)	X101	85.4
<b>Single-frame Detector + TROI</b>	X101	<b>82.3</b>
<b>SELSA (Wu et al. 2019) + TROI</b>	X101	<b>84.3</b>

Table 6: Performance comparison with other state-of-the-art models on the ImageNet VID validation set. <sup>△</sup> denotes using heavy video post-processing methods like Seq-NMS (Han et al. 2016). TROI denotes the proposed Temporal ROI Align

a stronger ResNeXt-101 backbone, Temporal ROI Align applied to SELSA achieves 84.3 mAP, which is the top one among existing methods that also adopt ResNeXt-101 backbone and do not use any post-processing techniques. Besides, our method uses less support frames (14 frames) than MEGA (35 frames), and the pipeline of our method is more simple than MEGA.

### Additional Experiments On EPIC KITCHENS

ImageNet VID dataset falls short in the density and diversity of objects. Therefore we evaluate Temporal ROI Align on the EPIC KITCHENS dataset (Damen et al. 2018). EPIC contains 290 classes and is far more complex and challenging. 272 video sequences captured in 28 kitchens are used for training. 106 sequences collected in the same 28 kitchens (S1) and 54 sequences collected in other 4 unseen kitchens (S2) are used for evaluation.

The results is shown in Table 7. Note that the performance of SELSA-*ReIm* (38.8 and 36.7) is higher than SELSA (Wu et al. 2019) (38.0 and 34.8). When SELSA-*ReIm* is equipped with the proposed Temporal ROI Align,

Method	mAP@0.5 (S1)	mAP@0.5 (S2)
SELSA (Wu et al. 2019)	38.0	34.8
SELSA- <i>ReIm</i>	38.8	36.7
SELSA- <i>ReIm</i> + TROI	<b>42.2</b>	<b>39.6</b>

Table 7: Performance comparison on EPIC KITCHENS test set. S1 and S2 indicate Seen and Unseen splits, respectively. SELSA-*ReIm* denotes the reimplementation of SELSA. TROI denotes Temporal ROI Align

Method	Backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>
MT R-CNN	R50-FPN	30.3	51.1	32.6
MT-CNN + TROI	R50-FPN	<b>33.5</b>	<b>57.0</b>	<b>36.6</b>
MT R-CNN	X101-FPN	34.9	58.8	36.5
MT R-CNN + TROI	X101-FPN	<b>38.0</b>	<b>63.3</b>	<b>40.3</b>

Table 8: Applying the Temporal ROI Align to MaskTrack R-CNN (MT R-CNN) in VIS. R50 denotes ResNet-50. AP denotes mask AP which follows the COCO evaluation metric to use 10 IoU thresholds from 50% to 95% at step 5%

the mAP@0.5 further improves to 42.2 and 39.6 by 3.4 and 2.9 points, respectively. This demonstrates that Temporal ROI Align can bring more performance gains on more complex video objection dataset compared with ImageNet VID.

### Extension: Experiments On VIS

We also investigate Temporal ROI Align on video instance segmentation (VIS). VIS aims at simultaneously detecting, segmenting and tracking object instances in a video sequence. The dataset of VIS is YouTube-VIS (Yang, Fan, and Xu 2019) which contains 40 object categories. It contains 2238 training videos, 302 validation videos, and 343 test videos. The training is performed on the training videos. Since the evaluation on the test set is currently closed, the evaluation is performed on the validation set.

MaskTrack R-CNN (Yang, Fan, and Xu 2019) is based on Mask R-CNN, and builds another branch (track head) to link the same object instances between two frames. MaskTrack R-CNN uses the conventional ROI Align to extract features from single-frame for proposals, and we replace the ROI Align with the proposed Temporal ROI Align. The results are shown in Table 8. We can see that the Temporal ROI Align consistently improves ResNet-50-FPN and ResNeXt-101-FPN baselines on all metrics involving AP, AP<sub>50</sub> and AP<sub>75</sub> (e.g. 3.2 points in AP of ResNet-50-FPN and 3.1 points in AP of ResNeXt-101-FPN), which further demonstrates the flexibility of proposed Temporal ROI Align.

### Visualization and Analysis

We visualize the most similar positions found by MS ROI Align and the corresponding averaged attention weights across different temporal attention blocks in TAFA.

Fig. 6 (a) shows the object-related position in the  $7 \times 7$  ROI features of the proposal. We can see that the red points are localized in the region of the same object instance, and the proposed TAFA can learn reasonable attention weights.



Figure 6: An illustration of the most similar positions found by MS ROI Align and the corresponding averaged attention weights across different temporal attention blocks in TAFE (zoom in for a better view). The support frames  $T$  is set to 4 during inference for visualization. The first column is the target frames and other columns are support frames. The dash green box is a proposal in target frames. The solid green box is the region projected into the target frame from the object-related position (a) or background-related position (b) in the  $7 \times 7$  ROI features of the proposal. The red points in support frames are the most similar positions corresponding to the object-related position (a) or background-related position (b) of the  $7 \times 7$  ROI features in target frame. Only the top 1 similar position rather than top 4 similar positions is visualized for a clear visualization. The floating numbers below each row of images are the averaged attention weights in TAFE corresponding to the object-related position (a) or background-related position (b) in the  $7 \times 7$  ROI features of the proposal



For example, in the third row of Fig. 6 (a), the objects are hard to be classified in the target frames due to motion blur. However, the object is clear in some support frames (e.g. the 3-rd row, 3-rd column in Fig. 6 (a)) and the MS ROI Align successfully locates the object-related points (red points). The attention weights of points belonging to clear object (e.g. 0.247 in the 3-rd row, 3-rd column of Fig. 6 (a)) are also larger than attention weights of points belonging to blurry object (e.g. 0.067 in the 3-rd row, 1-st column of Fig. 6 (a)). After the most similar features from red points in the support frames being integrated with the original ROI features by the reasonable attention weights in TAFA, it is much easier for the classifier to make the correct classification.

Fig. 6 (b) shows the background-related position in the  $7 \times 7$  ROI features of the proposal. We can see that most red points are localized in the surrounding region of the same object instance (e.g. the 1-rd row, 3-rd column in Fig. 6 (b)), which can be viewed as the context information of the object instance. If the red points are localized in the background region which is far away from object instances (e.g. the 1-rd row, 5-th column in Fig. 6 (b)), the corresponding attention weights in TAFA are very small (e.g. 0.032 in the 1-rd row, 5-rd column of Fig. 6 (b)), making the irrelevant background features less influence the original ROI features.

## Conclusion

In this paper, a novel Temporal ROI Align is proposed to offer an alternative for the conventional ROI Align in video object detection. Compared with ROI Align which only extracts features of proposals from single-frame, the proposed operator can extract features from the entire video for proposals, making the extracted features contain long-range temporal information and much easier to perform classification for region classifier. The proposed operator is flexible and also validated effectively on different baselines, different datasets and different vision tasks about videos. We hope the proposed operator be applied into other video-related tasks, such as multi object tracking, in the future.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. U20B2047, 62002336) and Exploration Fund Project of University of Science and Technology of China (YD3480002001).

## References

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

Cai, Z.; and Vasconcelos, N. 2018. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6154–6162.

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

*IEEE conference on computer vision and pattern recognition*, 7814–7823.

Chen, Y.; Cao, Y.; Hu, H.; and Wang, L. 2020. Memory Enhanced Global-Local Aggregation for Video Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dai, J.; Li, Y.; He, K.; and Sun, J. 2016. R-fcn: Object detection via region-based fully convolutional networks. In *Proceedings of the Advances in Neural Information Processing Systems*, 379–387.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 764–773.

Damen, D.; Doughty, H.; Maria Farinella, G.; Fidler, S.; Furnari, A.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; et al. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 720–736.

Deng, H.; Hua, Y.; Song, T.; Zhang, Z.; Xue, Z.; Ma, R.; Robertson, N.; and Guan, H. 2019a. Object Guided External Memory Network for Video Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 6678–6687.

Deng, J.; Pan, Y.; Yao, T.; Zhou, W.; Li, H.; and Mei, T. 2019b. Relation Distillation Networks for Video Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 7023–7032.

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

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 6569–6578.

Feichtenhofer, C.; Pinz, A.; Zisserman; and Andrew. 2017. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision*, 3038–3046.

Girshick, R. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Gu, J.; Hu, H.; Wang, L.; Wei, Y.; and Dai, J. 2018. Learning region features for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 381–395.

Guo, C.; Fan, B.; Gu, J.; Zhang, Q.; Xiang, S.; Prinet, V.; and Pan, C. 2019. Progressive Sparse Local Attention for Video

- Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 3909–3918.
- Han, W.; Khorrani, P.; Paine, T. L.; Ramachandran, P.; Babaeizadeh, M.; Shi, H.; Li, J.; Yan, S.; and Huang, T. S. 2016. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.
- Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3588–3597.
- Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.
- Liu, M.; and Zhu, M. 2018. Mobile video object detection with temporally-aware feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5686–5695.
- Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the Advances in Neural Information Processing Systems*, 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. *International journal of computer vision* 115(3): 211–252.
- Shvets, M.; Liu, W.; Berg, A. C.; and A. 2019. Leveraging Long-Range Temporal Relationships Between Proposals for Video Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 9756–9764.
- Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 9627–9636.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- Wang, S.; Zhou, Y.; Yan, J.; and Deng, Z. 2018a. Fully motion-aware network for video object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 542–557.
- Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.
- Wu, H.; Chen, Y.; Wang, N.; and Zhang, Z. 2019. Sequence Level Semantics Aggregation for Video Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 9217–9225.
- Xiao, F.; and Jae Lee, Y. 2018. Video object detection with an aligned spatial-temporal memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 485–501.
- Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.
- Yang, L.; Fan, Y.; and Xu, N. 2019. Video instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 5188–5197.
- Zhu, X.; Dai, J.; Yuan, L.; and Wei, Y. 2018. Towards high performance video object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7210–7218.
- Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9308–9316.
- Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017a. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 408–417.
- Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; and Wei, Y. 2017b. Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2349–2358.