Object Detection in Videos by Short and Long Range Object Linking

Peng Tang[†][*] Chunyu Wang[‡] Xinggang Wang[†] Wenyu Liu[†] Wenjun Zeng[‡] Jingdong Wang[‡]
[†]School of EIC, Huazhong University of Science and Technology [‡]Microsoft Research Asia
{pengtang,xgwang,liuwy}@hust.edu.cn {chnuwa,wezeng,jingdw}@microsoft.com

Abstract

We address the problem of detecting objects in videos with the interest in exploring temporal contexts. Our core idea is to link objects in the short and long ranges for improving the classification quality. Our approach first proposes a set of candidate spatio-temporal cuboids, each of which serves as a container associating the object across short range frames, for a short video segment. It then regresses the precise box locations in each frame over each cuboid proposal, yielding a tubelet with a single classification score which is aggregated from the scores of the boxes in the tubelet. Third, we extend the non-maximum suppression algorithm to remove spatially-overlapping tubelets in the short segment, avoiding tubelets broken by the frame-wise NMS. Finally, we link the tubelets across temporally-overlapping short segments over the whole video, in order to boost the classification scores for positive detections by aggregating the scores in the linked tubelets. Experiments on the ImageNet VID dataset shows that our approach achieves the state-of-the-art performance.

1. Introduction

Detecting objects in static images [2, 5, 6, 7, 23, 26, 27, 30] has achieved significant progress due to the emergence of deep convolutional neural networks [12, 19, 20, 32]. However, object detection in videos brings additional challenges such as motion blurs and degraded image qualities which may result in unstable classification for the same object across video. Therefore, many research efforts have been allocated to video object detection especially after the introduction of the ImageNet video object detection challenge (VID). Various attempts have been made to explore temporal contexts such as optical-flow based feature propagation [38] and aggregation [37], object association across frames by tubelets [16, 17, 18], and track to detect [4].

We propose a novel approach to explore the temporal contexts for video object detection by linking the objects in the short and long ranges and aggregating the classification scores from the linked objects in order to improve the classification quality. Figure 1 provides examples showing the benefits from short range and long range object linking.

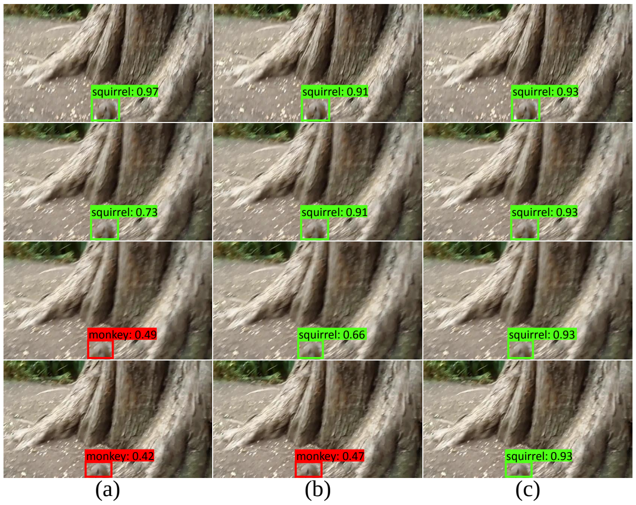

Figure 1. (a) Static image detections: Only the detections in the first two frames are correct. (b) Object tubelet detection (short range): The detection in third frame becomes correct due to object linking between the second and third frame. (c) Tubelet linking (long range): The detections in all the frames are correct due to tubelet linking. For each frame, we only show the top-scoring box, where green/red boxes correspond to success/failure examples.

We present a spatio-temporal cuboid proposal network to generate candidate spatio-temporal cuboids for a short video segment. The objects across frames lying in a cuboid are regarded as the same object. So the objects in a video segment are naturally linked by the cuboid. This differs from the existing approaches [4, 16] which link objects by predicting object movements between neighboring frames. In addition, our approach can be easily implemented by generalizing the region proposal network [27] with multiple frames as inputs.

On the basis of the cuboid proposals, we regress the precise box locations and classification scores for each frame separately, forming a tubelet representing the linked object boxes in the short video segment. We obtain the classifi-

[*]This work was done during Microsoft Research Asia internship.

cation score of the tubelet, by aggregating the classification scores of the boxes across frames. We then extend the standard non-maximum suppression (NMS) to remove overlapping tubelets with the introduction of an overlap measure for tubelets, which prevents the tubelets from breaking that may happen in frame-wise NMS.

Finally, we link the tubelets with significant overlap across temporally-overlapping short video segments that are obtained by dividing the whole video sequence, yielding long range object linking. If two boxes, which are from the *temporally-overlapping* frame of two neighboring tubelets, have sufficient spatial overlap, the tubelets are linked together and merged. This differs from tracklet linking [4] that links object boxes across *non-overlapping* frames. The benefit of linking through the overlapping frame is that there is no need to predict the movement between neighboring frames which might not be accurate enough. We exploit the long range object linking to improve the classification quality by boosting the classification scores for positive detections through aggregating the classification scores of the linked tubelets. Our approach achieves the state-of-the art performance on the VID dataset [28]: the mAP score is 74.3% only using the VID for training and is 80.6% using both the VID and DET for training.

2. Related work

The task of object detection in both images [2, 5, 6, 7, 10, 22, 23, 34] and videos [4, 16, 17, 37, 38] has been widely studied in the literature. We mainly review related works on video object detection and classify them into three categories by how they use the temporal contexts.

Feature propagation w/o object association. In [37], the features of the current frame are augmented by aggregating the features propagated from the neighboring frames based on the pixel-wise correspondences which are established by computing the optical flows [3]. Feature propagation is also exploited in [38] to speed up the object detections. The authors propose to compute the feature maps (using a very deep network with high computation cost) for the key frames and propagate the features to non-key frames by computing the optical flows using a shallow network which takes less time. These methods are different from ours because they do not perform object associations[1].

Feature propagation w/ object association. The tubelet proposal network approach [16] computes the tubelet by using the same proposal computed from the first frame, for connecting the detection boxes over a sequence of frames. The features of the boxes in the tubelets are propagated to each box for classification by using a CNN-LSTM classification network. The main difference between our work and [16] is that we propagate boxes scores instead of features across frames. Besides, we directly generate the spatio-temporal cuboid proposals for video segments rather than per-frame proposals in [16].

Score propagation w/ object association. The approach in [17, 18] proposes two kinds of object associations. The first one tracks the detected box in current frame to its neighboring frames to augment their original detections, which increases the object recall. The scores are also propagated to improve classification accuracy. The association is based on the mean optical flow vector within boxes. The second one associates objects into long tubelets using the tracking algorithm [35] and then adopts a classifier to aggregate the detection scores in the associated tubelets. The Seq-NMS approach [9] builds object associations by checking the overlap between boxes of the neighboring frames without considering the motion information and then aggregates the scores of the associated objects for the final score. The *detect to track and track to detect* approach [4] simultaneously predicts the object locations in two frames and also the displacement from the preceding frame to the current frame. Then they use the displacement to link the detected objects into class-specific tubelets. The object detection scores in the same tubelet is reweighed by aggregating the scores in some manner from the scores in that tubelet.

Our approach belongs to the third category. Different from previous methods, we associate the detected objects in two levels. (1) Short range object linking: we learn the spatio-temporal cuboid proposals to detect object tubelets, a representation of linked objects across frames, where the objects in the same tubelet are about the same object with a single classification score, which enables us to generate temporally-overlapping tubelets for the long range linking. (2) Long range object linking: we link temporally-overlapping class-specific tubelets and aggregate their classification scores as the score for the linked tubelet, which boosts classification scores of positive detections.

3. Our approach

The task of video object detection infers the locations and the classes of the objects in each frame of a video $\{\mathbf{I}^1, \mathbf{I}^2, \dots, \mathbf{I}^N\}$. Our approach exploits the object associations across frames for robust object detection and consists of four stages, three about short range object linking and one about long range object linking: (1) Cuboid proposal for a short video segment. This step aims to propose a set of cuboids (containers) which bound the same object across frames. See Figure 2. (2) Tubelet detection for a short segment. Given the cuboid proposals, the goal is to regress and classify a tubelet which is a sequence of bounding boxes with each box localizing the object in one frame. The tubelet is a representation for *linked objects across frames in a short segment*, which is illustrated in

[1] "Association" and "linking" are interchangeable in this paper.

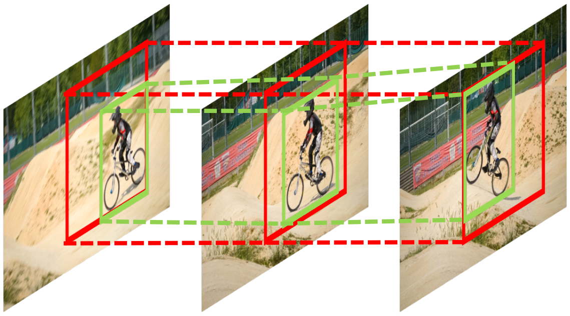

Figure 2. The red cuboid, bounding the movement of the object, is the target of the *cuboid proposal* stage. The tubelet, composed of the green object boxes in the video segment, is the target of the *object tubelet objection* stage.

Figure 2. (3) Tubelet non-maximum suppression. It is used to remove spatially-overlapping tubelets. (4) Classification refinement through temporally-overlapping tubelets. This step, depicted in Figure 3, links the *temporally-overlapping tubelets* to associate objects in the whole sequence, and refines the classification scores of the linked tubelets.

3.1. Cuboid proposal for a short segment

The ground truth bounding cuboid of the objects in a short segment, containing K frames $\{\mathbf{I}^t, \mathbf{I}^{t+1}, \dots, \mathbf{I}^{t+K-1}\}$, is defined as follows. Let $\tilde{\mathbf{b}}$ be the $2D$ bounding box of the tubelet $\tilde{\mathcal{T}} = (\tilde{\mathbf{b}}^t, \tilde{\mathbf{b}}^{t+1}, \dots, \tilde{\mathbf{b}}^{t+K-1})$, a series of all ground truth boxes in the K frames,

$$\tilde{\mathbf{b}} = \mathrm{BoundingBox}(\tilde{\mathbf{b}}^t, \tilde{\mathbf{b}}^{t+1}, \dots, \tilde{\mathbf{b}}^{t+K-1}). \quad (1)$$

Here, $\tilde{\mathbf{b}}^\tau = (x^\tau, y^\tau, w^\tau, h^\tau)$ in frame τ, denoting the horizontal and vertical center coordinates and its width and height, is the ground truth box of frame τ. The bounding cuboid in our approach is just a collection of K $\tilde{\mathbf{b}}$s: $\tilde{\mathbf{c}} = (\tilde{\mathbf{b}}, \tilde{\mathbf{b}}, \dots, \tilde{\mathbf{b}})$, and thus simplified as a $2D$ box $\tilde{\mathbf{b}}$. Figure 2 provides the examples about the cuboid and the tubelet in a short segment.

We modify the region proposal network (RPN) approach in Faster R-CNN [27] and introduce the *cuboid proposal network* (CPN) approach for computing cuboid proposals. Unlike the conventional RPN where the input is usually a single image, our approach takes the K frames as the input to the CPN. The output is a set of whk candidate cuboids of interest (CoI), regressed from a $w \times h$ spatial grid, where there are k reference boxes at each location, and each CoI is associated with a objectness score.

3.2. Object tubelet detection for a short segment

We use the $2D$ form of the cuboid proposal as the $2D$ box (region) proposal for each frame in this segment, which is classified and refined for each frame separately.

Considering a frame $\mathbf{I}^\tau$ in this segment, we follow Fast R-CNN [6] to refine the box and compute the classification score. We start with a RoI pooling operation, where the input is a $2D$ region proposal $\mathbf{b}$ and the response map of $\mathbf{I}^\tau$ obtained through a CNN. The RoI pooling result is fed into a classification layer, outputting a $\{C+1\}$-dimensional classification score vector $\mathbf{y}^\tau$, where C is the number of categories and 1 corresponds to the background, as well as a regression layer, from which the refined box is obtained.

The resulting K refined boxes for the K frames form the tubelet detection result over this segment, $\mathcal{T} = (\mathbf{b}^t, \mathbf{b}^{t+1}, \dots, \mathbf{b}^{t+K-1})$. The classification score of this tubelet is an aggregation of the scores over all the frames,

$$\bar{\mathbf{y}} = \mathrm{aggregation}(\mathbf{y}^t, \mathbf{y}^{t+1}, \dots, \mathbf{y}^{t+K-1}), \quad (2)$$

where $\mathrm{aggregation}(\cdot)$ could be a mean operation. We empirically find that $\mathrm{Aggregation}(\cdot) = \frac{1}{2}(\mathrm{mean}(\cdot) + \max(\cdot))$ performs the best.

3.3. Tubelet non-maximum suppression

The non-maximum suppression (NMS) algorithm aims to remove overlapping object detection boxes. It sorts all the detected boxes on the basis of their scores. The detection box with maximum score is selected and all other detection boxes with a significant overlap (larger than a predefined threshold) with the selected boxes are suppressed. This process is recursively applied to the remaining boxes.

The straightforward solution is to conduct frame-wise NMS: remove overlapping 2D boxes independently for each frame, which tends to break a tubelet into small tubelets that are classified to different objects because the point that the objects in a tubelet are about the same object is not exploited. The empirical results also verify this tendency.

Instead, we extend the NMS algorithm to remove overlapping object tubelets, resulting a tubelet NMS (T-NMS) algorithm. The main point lies in how to measure the overlap between two tubelets. We define it on the base of the overlap between the boxes in the same frame. Given two tubelets, $\mathcal{T}_i = (\mathbf{b}_i^t, \mathbf{b}_i^{t+1}, \dots, \mathbf{b}_i^{t+K-1})$ and $\mathcal{T}_j = (\mathbf{b}_j^t, \mathbf{b}_j^{t+1}, \dots, \mathbf{b}_j^{t+K-1})$, the overlap is computed as

$$\mathrm{overlap}(\mathcal{T}_i, \mathcal{T}_j) = \min_{\tau = t, t+1, \dots, t+K-1} \mathrm{IoU}(\mathbf{b}_i^\tau, \mathbf{b}_j^\tau), \quad (3)$$

where $\mathrm{IoU}(\mathbf{b}_i^\tau, \mathbf{b}_j^\tau)$ is the intersection over union between $\mathbf{b}_i^\tau$ and $\mathbf{b}_j^\tau$ for frame τ.

3.4. Classification score refinement via temporally-overlapping tubelets

Our approach divides the video with a series of overlapping short segments of length K with stride equal to $K-1$:

$$\mathcal{S}^1 = (\mathbf{I}^1, \mathbf{I}^2, \dots, \mathbf{I}^K), \quad (4)$$

$$\mathcal{S}^2 = (\mathbf{I}^K, \mathbf{I}^{K+1}, \dots, \mathbf{I}^{2K-1}), \quad (5)$$

$$\dots\dots \quad (6)$$

$$\mathcal{S}^M = (\mathbf{I}^{(M-1)K-M+2}, \dots, \mathbf{I}^{MK-M+1}). \quad (7)$$

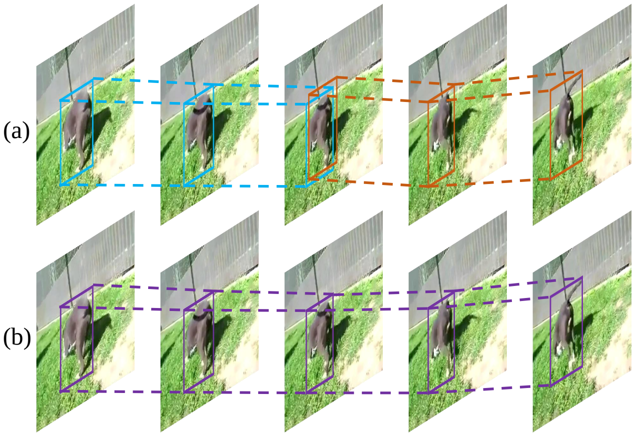

Figure 3. Illustration of long range linking. Boxes with the same color belong to the same tubelet. The short tubelets (a) from two neighboring video segments are linked together to form new long tubelets (b).

Considering two temporally-overlapping tubelets: the ith tubelet from the mth segment and the i'th tubelet from the $(m+1)$th segment:

$$\mathcal{T}_i^m = (\mathbf{b}_i^{t_m}, \mathbf{b}_i^{t_m+1}, \ldots, \mathbf{b}_i^{t_m+K-1}),$$

$$\mathcal{T}_{i'}^{m+1} = (\mathbf{b}_{i'}^{t_m+K-1}, \mathbf{b}_{i'}^{t_m+K}, \ldots, \mathbf{b}_{i'}^{t_m+2K-1}),$$

we link them if the overlap between $\mathbf{b}_i^{t_m+K-1}$ and $\mathbf{b}_{i'}^{t_m+K-1}$ from the overlapping frame is significant and larger than a pre-defined threshold.

We perform a greedy tubelet linking algorithm. Initially, we put the tubelets of all the short segments into a pool and record the corresponding segment for each tubelet. Our algorithm pops out the tubelet $\mathcal{T}$ with the highest classification score from the pool. We check the IoU of the boxes over the overlapping frame between $\mathcal{T}$ and its temporally-overlapping tubelets. If the IoU is larger than a threshold, fixed as 0.4 in our implementation, we merge the two tubelets into a single tubelet, remove the box with the lower score for the overlapping frame, update the classification score for the merged tubelet according to Equation (2), and record the corresponding segment (a combination of the corresponding two video segments). We then push the merged tubelet into the pool. This process is repeated until no more tubelets can be merged. Figure 3 gives the examples of linking short tubelets to form long range linking results.

The tubelets remaining in the pool form the video object detection results: the score of the tubelet is assigned to each box in the tubelet, and the boxes from the all the tubelets associated with a frame are regarded as the final detection boxes for the corresponding frame.

3.5. Implementation details

Cuboid proposal. The base network is ResNet-101 [12] pre-trained on the ILSVRC dataset [28]: we remove all layers after the *Res5c* layer and replace the convolutional layers in the fifth block by dilated ones [1, 36] to reduce the stride from 32 to 16. On the basis of the base network, we add a convolutional layer with 512 filters of 3×3, and use two convolutional layers of 1×1 to regress the offsets and predict the objectness scores for cuboid proposals. The network is split into two sub-networks: the first one has two residual blocks pass each frame separately to obtain frame-specific features which are concatenated as input of the second sub-network with three residual blocks.

We use four anchor scales 64^2, 128^2, 256^2, and 512^2 with three aspect ratios $1:1$, $1:2$, and $2:1$, resulting in 12 anchors at each location in total. The length K of each video segment will be studied in our experiments. The loss function is the same to that in the standard RPN [27]: the cross-entropy loss for classification and the smoothed L1 loss for regression. The NMS threshold 0.7 is chosen and at most 300 proposals are kept for the detection network training/testing. In the testing stage, if the number of frames in the last segment is smaller than K, we pad the segment by some frames copied from the last frame.

Object tubelet detection. The base network is the same as it for cuboid proposal. We use RoI pooling to extract 7×7 response maps from the layer *Res5c*, followed by two fully-connected + ReLU layers (1024 neurons), outputting the features. We use one fully connected layer for classification, and another fully connected layer for bounding box regression. Following the Fast R-CNN [6], we train the network with online hard example mining method [31]. During testing, the T-NMS threshold is set to 0.4.

Training. We use the SGD algorithm to train the cuboid proposal network and the object tubelet detection network. We initialize the base networks from the pretrainded model, and the weights in the layers added on top of the *Res5c layer* in the base network using a zero-mean Gaussian distribution whose std is 0.01. The training images are resized such that their shorter side is 600 pixels (The same process is conducted during the testing stage). We set the mini-batch size to 8, the learning rate to 1×10^{-3} for the first 40K iterations and 1×10^{-4} for the next 20K iterations, and the momentum to 0.9. We do not find the gain from sharing the base networks for the cuboid proposal network and the object tubelet detection network and thus simply train them separately. Our implementation is based on the Caffe [14] deep learning framework on a NVIDIA TitanX (Pascal) GPU.

3.6. Discussions

Cuboid and tubelet. The concept, tubelet, was studied in [16] for video object detection. The tubelet, though called proposal in [16], is in some sense similar to the tubelet defined in our approach. The key difference lies in that our approach uses the tubelet as an object association representation: the boxes in that tubelet are about the same object,

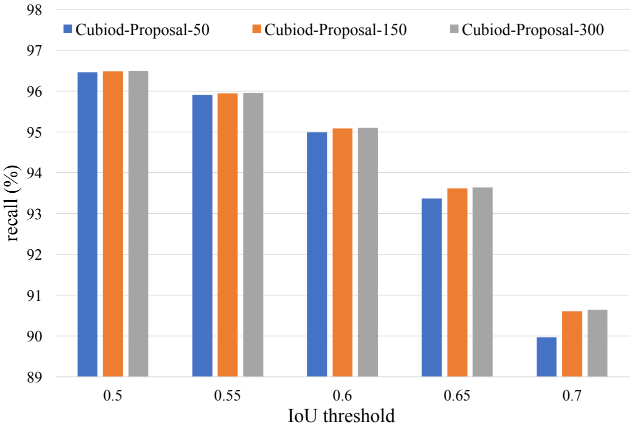

Figure 4. Recall *vs*. IoU threshold on the VID validation set. We show the results when keeping 50, 150, and 300 cuboid proposals for IoU threshold 0.5 to 0.7.

while the tubelet [16] is used for aggregating the features across the boxes in that tubelet and propagating the aggregated feature to each box for classification. The static object proposal computed from the first frame [16] is used as static object proposals for the later frames in the video segment, which is actually like the cuboid proposal presented in our approach. Differently, our approach computes the cuboid proposal from all the frames in the segment.

Tubelet linking. Our proposed tubelet linking approach is close to tubelet/tracklet linking introduced in [4, 17] for helping video object detection. There is a clear difference: our approach links two tubelets by check if the boxes for the *same* frame in two tubelets are about the same object, while the approaches in [4, 17] check if the boxes of the *neighboring* frames are linked through the tracking technoque, *e.g*., the regressed movement across frames [4] or FCN tracker [35].

Action detection. After the paper submission, the reviewers pointed out that there are some related works on action detection [8, 13, 15, 25, 29, 33]. Similar to [4, 17], the approaches in [8, 13, 25, 29, 33] link objects through *neighboring* frames. The concurrent work [15] is most similar to ours. But video object detection and action detection are different tasks: classification in video object detection can be done for single frames, while classification in action is basically a problem of temporal process classification. Therefore, the similar components have different usage/observations, *e.g*., different short video segment lengths and detecting objects in different frames separately/jointly.

Boundary issue. It is possible that in a video segment, an object may not appear in all the frames. We investigate the VID dataset and find that this boundary issue only leads to small performance drop (up to 0.57%). As a future work, we will adopt the 1D belief propagation algorithm to optimally re-label the boxes for each frame in the linked tubelets to resolve the bounding issue.

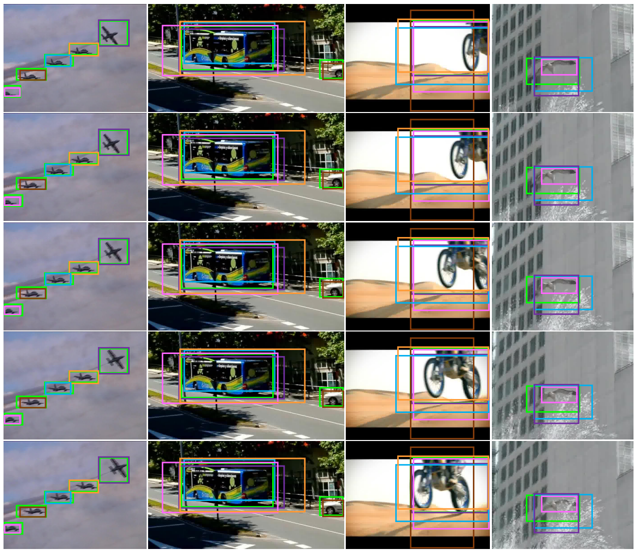

Figure 5. Visualization of some cuboid proposal results. Each column corresponds to a short video segment. The green boxes are the groundtruths and the rest are the cuboid proposals. Boxes with the same color belong to the same cuboid proposals. For each video segment, we only show five proposals with the highest objectness scores for simplicity.

4. Experiments

4.1. Dataset and evaluation metric

We use the ImageNet video object detection (VID) dataset [28] which was introduced in the ILSVRC 2015 challenge. There are 30 object classes in the dataset which cover different movement types and different levels of clutterness. The dataset has 5354 videos in total which are divided into the training, validation, and testing subsets with 3862, 555, and 937 videos, respectively. Each video has about 300 frames on average. The dataset provides groundtruth object locations, labels, and object identifications for each frame of the video. Since the annotations for the testing subset has been reserved for the challenge and the evaluation server has been closed, we test on the validation subset as most of the other works.

The VID dataset provides two evaluation metrics: one is the classical detection evaluation metric, *i.e*., the Average Percision (AP) and mean of AP (mAP) over all classes, which treats boxes in each frame independently during evaluation; the another one is a new tracking metric which evaluates both detection accuracy in each frame and also the linking accuracy over the whole video. We mainly compare our performance with previous methods focusing on the detection [4, 16, 17, 18, 37, 38]. Besides, we also choose the tracking metric to show that our method can be further applied to tracking.

Table 1. Detection results (mAP in %) of different methods on the VID validation set which is divided into three subsets according to the object moving speed. The relative gains over the static image detector baseline (a) are listed in the subscript. Here the video segment length is set to 2 for (b-d).

Methods	(a)	(b)	(c)	(d)
CPN		✓	✓	✓
T-NMS			✓	✓
Linking				✓
mAP (%)	69.1	$69.3_{\uparrow 0.2}$	$70.5_{\uparrow 1.4}$	$\mathbf{74.3_{\uparrow 5.2}}$
mAP (%) (slow)	76.8	$76.7_{\downarrow 0.1}$	$77.7_{\uparrow 0.9}$	$\mathbf{80.5_{\uparrow 3.7}}$
mAP (%) (medium)	68.5	$68.9_{\uparrow 0.4}$	$70.2_{\uparrow 1.7}$	$\mathbf{74.5_{\uparrow 6.0}}$
mAP (%) (fast)	47.2	$47.7_{\uparrow 0.5}$	$49.4_{\uparrow 2.2}$	$\mathbf{55.1_{\uparrow 7.9}}$

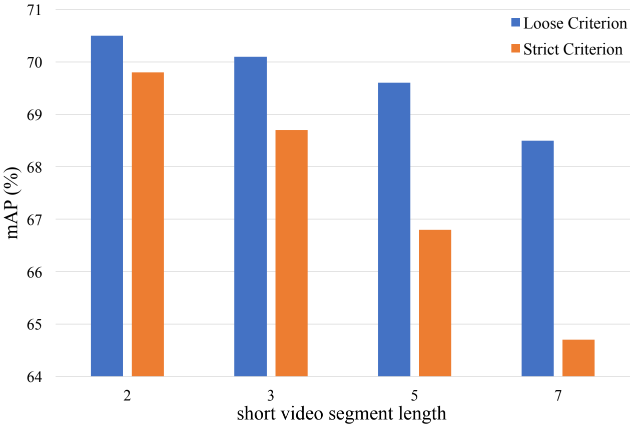

Figure 6. Short range linking results (mAP in %). The more strict evaluation criterion checks whether the instance IDs of the boxes in a tubelet are the same.

4.2. Ablation studies

We first conduct detailed ablation experiments to study the effectiveness of different settings in our method, including cuboid proposal recall, short range linking, video segment length, NMS/T-NMS, and long range linking. In particular, the static image detector baseline mentioned below is a Faster R-CNN network [27] that treats all frames as static images without considering temporal information.

Cuboid proposal recall. We first evaluate the recall of proposals by CPN. To do this, we generate a collection of cuboid proposals for each video segment and compute their recall at different IoU thresholds (0.5 to 0.7) with groundtruth cuboids. Figure 4 shows the quantitative results on the validation set. Firstly, we can see that keeping as few as 50 proposals already gives reasonably good performance: more than 96.46% of the groundtruths are recalled when the IoU threshold is set to 0.5. Secondly, increasing the number of proposals brings only marginal gains for lower IoU thresholds (*e.g.* 0.5) and gives larger gains for higher IoU thresholds (*e.g.* 0.7). The results show that choosing 300 proposals can already achieve satisfactory recall. Thus we only use 300 proposals for following experiments. We also show several qualitative results in Figure 5. The green boxes are the groundtruth and the rest are the proposals generated by CPN. In most cases, there is at least one proposal that has sufficient overlap with the groundtruths, which shows that the CPN can accurately learn the boundaries of the object movement. Take the aeroplane as an example, the boxes in the top figure touch the aeroplane tails while those in the bottom touch the aeroplane heads. Besides, it deals well with the videos having single/multiple, small/large, fast/slow moving objects.

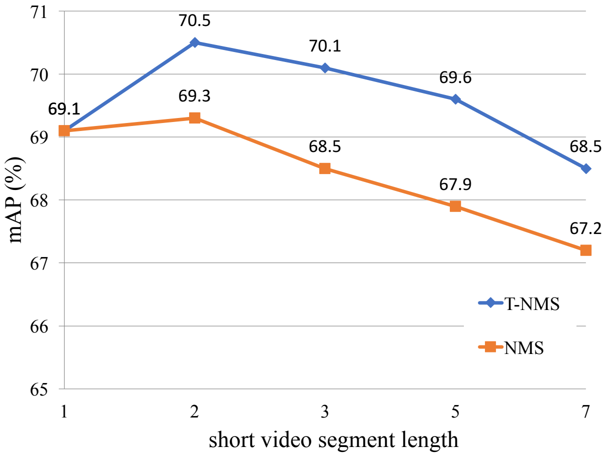

Figure 7. Detection results (mAP in %) for NMS/T-NMS and different short video segment lengths, where video segment length 1 means the static image detector.

Short range object linking. We then investigate whether the boxes in the tubelets for short segments correspond to the same objects. For a testing video, our method first generates a set of tubelets. Then if all boxes in the tubelet localize object accurately and correspond to the same object, the tubelet is be classified as true positive, and otherwise it is a false positive. After that we can compute precision-recall curves and obtain mAP. It is obvious that this is a more strict evaluation criterion than the one used for video object detection. Figure 6 shows the results. We can see that using the strict evaluation protocol only slightly decreases the performance (*e.g.*, from 70.5% to 69.8% or from 70.1% to 68.7%), which justifies that the linking results of our method are reasonably accurate.

The influence of short video segment length. We discuss the influence of short video segment length . From Figure 7, we can see that using video segment lengths of 2, 3 and 5 (with T-NMS) can all improve over the static baseline. The largest improvement (1.4% mAP) is obtained when the video segment length is 2. When the video segment length increases, the performance decreases. In addition, as shown in Figure 6, we can see that the short range linking performance for long video segments is worse than short ones. There are several reasons explaining this phenomenon. First, longer segments are more probable to generate oversized proposals which have smaller overlap with the groundtruth boxes in each frame. Second, the oversized

Table 2. Average precision (in %) for different methods on the VID dataset. ⋆ indicates models trained on the mixture of VID and DET datasets.

Method	aero	antelope	bear	bike	bird	bus	car	cattle	dog	cat	elephant	fox	g_panda	hamster	horse	lion	lizard	monkey	mbike	rabbit	r_panda	sheep	snake	squirrel	tiger	train	turtle	boat	whale	zebra	mAP
Kang *et al.* [16]	84.6	78.1	72.0	67.2	68.0	80.1	54.7	61.2	61.6	78.9	71.6	83.2	78.1	91.5	66.8	21.6	74.4	36.6	76.3	51.4	70.6	64.2	61.2	42.3	84.8	78.1	77.2	61.5	66.9	88.5	68.4
Kang *et al.* [17]⋆	83.7	**85.7**	84.4	74.5	73.8	75.7	57.1	58.7	72.3	69.2	80.2	83.4	80.5	93.1	**84.2**	67.8	80.3	54.8	80.6	63.7	85.7	60.5	72.9	52.7	89.7	81.3	73.7	69.5	33.5	90.2	73.8
Lee *et al.* [21]⋆	86.3	83.4	88.2	**78.9**	65.9	**90.6**	**66.3**	**81.5**	72.1	76.8	82.4	88.9	**91.3**	89.3	66.5	38.0	77.1	57.3	**88.8**	78.2	77.7	40.6	50.3	44.3	91.8	78.2	75.1	**81.7**	63.1	85.2	74.5
Zhu *et al.* [37]⋆	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	78.4
Feichtenhofer *et al.* [4]⋆	90.2	82.3	87.9	70.1	73.2	87.7	57.0	80.6	77.3	82.6	**83.0**	**97.8**	85.8	96.6	82.1	66.7	**83.4**	**57.6**	86.7	74.2	**91.6**	59.7	76.4	**68.4**	**92.6**	**86.1**	**84.3**	69.7	66.3	**95.2**	79.8
Ours	89.5	77.5	81.1	71.3	71.8	84.9	60.1	69.9	68.7	85.2	79.7	90.0	83.7	93.5	66.5	46.5	74.7	48.9	80.6	56.1	74.8	66.7	60.7	48.3	91.1	84.6	82.4	73.3	75.1	91.2	74.3
Ours⋆	**90.5**	80.1	**89.0**	75.7	**75.5**	83.5	64.0	71.4	**81.3**	**92.3**	80.0	96.1	87.6	**97.8**	77.5	**73.1**	81.5	56.0	85.7	**79.9**	87.0	**68.8**	**80.7**	61.6	91.6	85.5	81.3	73.6	**77.4**	91.9	**80.6**

proposals are probable to overlap with the image regions of other objects which causes more ambiguities for accurate localization and classification. Due to the better detection and linking results, we set the video segment length to 2 in following experiments if not specified.

NMS *vs*. T-NMS. We also study the influence of NMS/T-NMS for object detection. The NMS is implemented by removing boxes for each frame separately instead of removing tubelets for video segments in the T-NMS. Figure 7 and Table 1 show that T-NMS gives consistently better performance than NMS, which demonstrates that comparing with NMS handle each frame separately, simply considering short range temporal contexts contributes to better detection results.

Long range linking. Finally, we show the improvement by long range linking. We evaluate the performance on slow, medium, and fast ones which are formed according to their speed as done in [37]. As we can see in Table 1, comparing with the static baseline, considering both short and long range temporal information can boost the performance. When linking objects over the whole video to consider long range temporal context, there is significant improvements (5.2% to static and 3.8% to short range linking). Importantly, the performance gains are mainly from the faster objects (6.0% for medium and 7.9% for fast). It is natural that faster objects may have more variations, thus detecting them depends more on temporal context. As long range linking performs much better than other, in the following we only report results by long range linking.

4.3. Results

We compare our detection results with the current state-of-the-arts in Table 2. First, when only training on the VID dataset, our method obtains the superior result 74.3% mAP. To pursue the state-of-the-art detection performance, we follow the previous methods [4, 17, 37, 38] to use the mixture of ImageNet VID and DET datasets for training the detection network, and utilize the standard multi-scale training and testing [11]. As we can see, comparing our 80.6% with other methods using the same ResNet-101 network [4, 37], our method also obtains better performance, which confirms the effectiveness of our linking strategy. In particular, Zhu *et al.* [37] achieve the 78.4% performance by combining feature propagation and the score propagation [9]. Feichtenhofer *et al.* [4] add a tracking loss to learn better features for performance improvement. There are potential benefits from learning better features in the proposal and detection stages by incorporating other methods such as feature propagation and extra losses into our method.

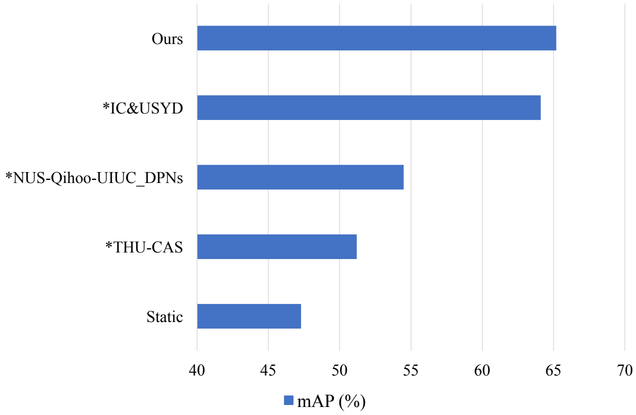

Figure 8. Tracking performance (mAP in %). * denotes results from http://image-net.org/challenges/LSVRC/2017/results reported on the VID test set.

Figure 9 visualizes several detection result comparisons between the static image detector and our method. From the first two rows, we can see that the static method fails to detect the red-panda when there are severe motion blurs and occlusions. This is reasonable because the appearance features have been severely degraded in this situation. After applying the short and long range linking and rescoring, our method successfully classifies the target in the challenging frames. In addition, it is common that the static detectors may confuse with similar classes (*e.g.*, bikes *vs*. motorbikes, cats *vs*. dogs) especially when a frame has low image quality. This problem can also be alleviated by rescoring the detections in the whole video because some frames have

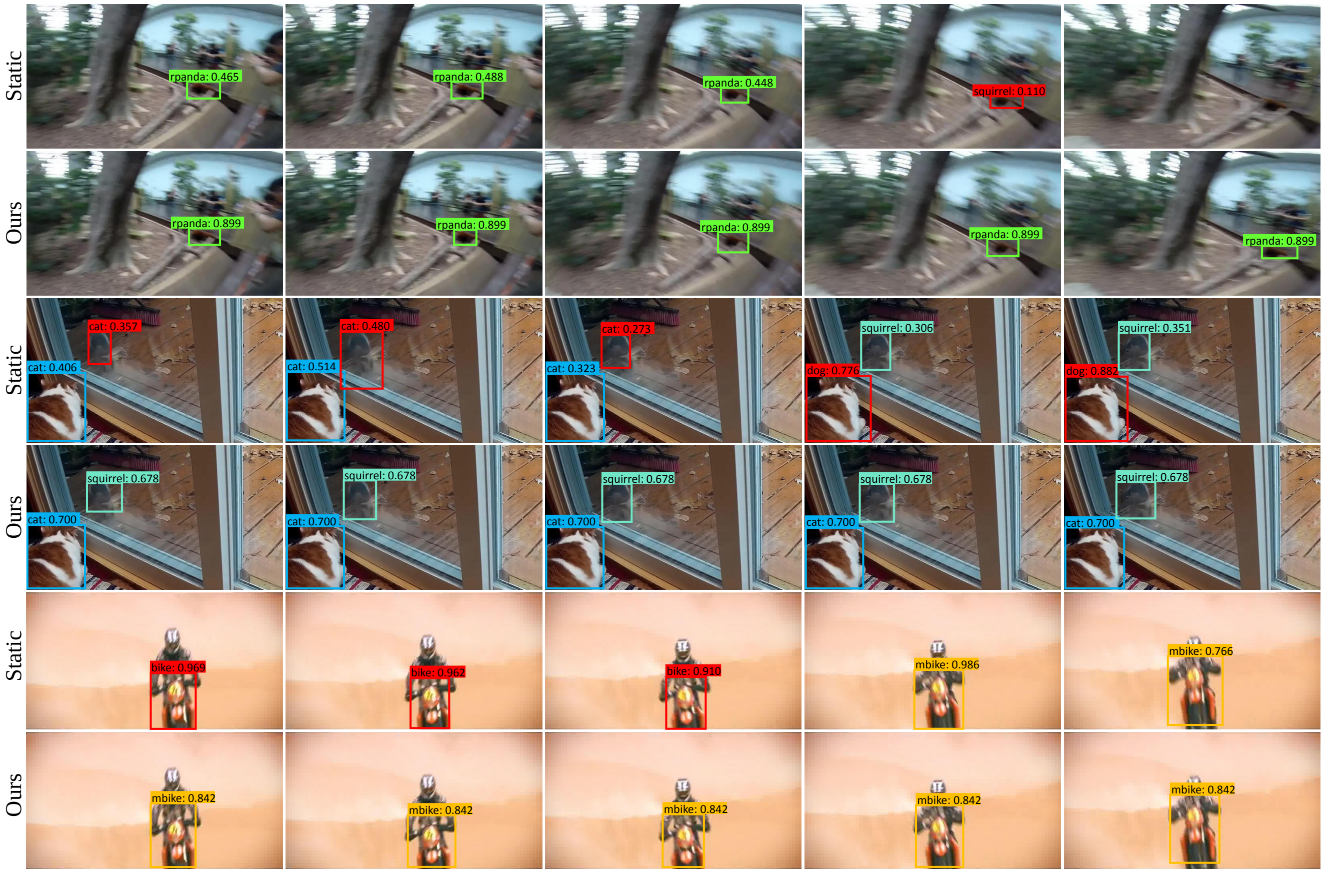

Figure 9. Example detection results of the static detector and our method (only the top-scoring boxes around objects are shown). Each row shows the results of sampled successive frames. For the linking results, boxes with the same color belong to the same tubelet in the video. Our method outperforms the static image detector when there are motion blurs, video defocus, and occlusions in the video.

correct classifications and can propagate these scores to the challenging frames by short and long range linking.

4.4. Application to tracking

We show that our long range linking results can be treated as object tracking results directly. The protocol in the ImageNet VID challenge [24] is used for evaluation. Specifically, for a testing video, our method first generates a set of linked detections. For each link, if all of its boxes have sufficient overlap (*i.e.* 0.5) with the corresponding groundtruth boxes, and meanwhile the temporal overlap between the link and the groundtruth is larger than a threshold, the link is regarded as a true positive. Otherwise, it is a false positive. The temporal thresholds are set to 0.25, 0.5, and 0.75. The higher the threshold is, the longer time we expect the method can successfully track.

Figure 8 shows the results. In detail, the mAPs are 68.2%, 65.1%, and 62.5% when the temporal overlap thresholds are set to 0.25, 0.5, and 0.75, respectively. The numbers suggest that our method can robustly generate long links. The method "Static" uses the static image detector and links the boxes by computing the overlap between neighboring frames. The results are much worse than ours, which confirms that linking objects in the overlapping frame is more preferable as it avoids to predict movement among neighboring frames. Since there are no published results on the VID validation set for other methods, we list the numbers on the test set of the challenge winners.

4.5. Runtime

For the case of two frame segments, our method takes 0.35s per-frame for testing which is comparable to 0.30s by the static baseline. The small extra cost comes from the cuboid proposal generation procedure: a small sub-network processing the two frames separately. The extra time cost is small for the detection stage due to the shared convolutional feature map, the computation time of T-NMS is almost the same as the NMS, and the long range linking is very efficient (about 1ms per-frame). The speed becomes even faster than the baseline when the short video segment length is larger than 2 (*e.g.*, 0.27s and 0.23s for video segment length 3 and 5 respectively), as the CPN generates cuboid proposals for all the frames in the segment by computing the features once.

5. Conclusion

We exploit objects association across frames in the video for improving the classification quality. The key factors that contribute to the superior performance of our proposed approach include: (1) Cuboid proposals for short range object linking; (2) An extension of NMS to tubelets with avoidance of broken tubelets; and (3) Classification score propagation across tubelets by (long range) tubelet linking. The resulting approach is simple, easily implemented. Our approach achieves the state-of-the-art video detection performance on the VID dataset.

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 4

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005. 1, 2

[3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015. 2

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *ICCV*, pages 3038–3046, 2017. 1, 2, 5, 7

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010. 1, 2

[6] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 1, 2, 3, 4

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1, 2

[8] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, pages 759–768, 2015. 5

[9] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang. Seq-nms for video object detection. *arXiv preprint arXiv:1602.08465*, 2016. 2, 7

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015. 7

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 4

[13] R. Hou, C. Chen, and M. Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *ICCV*, pages 5822–5831, 2017. 5

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678, 2014. 4

[15] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, pages 4405–4413, 2017. 5

[16] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang. Object detection in videos with tubelet proposal networks. In *CVPR*, pages 727–735, 2017. 1, 2, 4, 5, 7

[17] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *arXiv preprint arXiv:1604.02532*, 2016. 1, 2, 5, 7

[18] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, pages 817–825, 2016. 1, 2, 5

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[21] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee. Multi-class multi-object tracking using changing point detection. In *ECCV*, pages 68–83, 2016. 7

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 1, 2

[24] W. Liu, O. Russakovsky, J. Deng, F.-F. Li, and A. Berg. ILSVRC 2016 object detection from video. http://image-net.org/challenges/talks/2016/ILSVRC2016_10_09_vid.pdf, 2016. 8

[25] X. Peng and C. Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, pages 744–759, 2016. 5

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 3, 4, 6

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 2, 4, 5

[29] S. Saha, G. Singh, and F. Cuzzolin. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In *ICCV*, pages 4414–4423, 2017. 5

[30] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR*, volume 1, pages 746–751, 2000. 1

[31] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016. 4

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[33] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*, pages 3637–3646, 2017. 5

[34] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 2

[35] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, pages 3119–3127, 2015. 2, 5

[36] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 4

[37] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, pages 408–417, 2017. 1, 2, 5, 7

[38] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *CVPR*, pages 2349–2358, 2017. 1, 2, 5, 7