Object Detection from Video Tubelets with Convolutional Neural Networks

Kai Kang Wanli Ouyang Hongsheng Li Xiaogang Wang
Department of Electronic Engineering, The Chinese University of Hong Kong
{kkang,wlouyang,hsli,xgwang}@ee.cuhk.edu.hk

Abstract

*Deep Convolution Neural Networks (CNNs) have shown impressive performance in various vision tasks such as image classification, object detection and semantic segmentation. For object detection, particularly in still images, the performance has been significantly increased last year thanks to powerful deep networks (*e.g. *GoogleNet) and detection frameworks (*e.g. *Regions with CNN features (R-CNN)). The lately introduced ImageNet [6] task on object detection from video (VID) brings the object detection task into the video domain, in which objects' locations at each frame are required to be annotated with bounding boxes. In this work, we introduce a complete framework for the VID task based on still-image object detection and general object tracking. Their relations and contributions in the VID task are thoroughly studied and evaluated. In addition, a temporal convolution network is proposed to incorporate temporal information to regularize the detection results and shows its effectiveness for the task. Code is available at* https://github.com/myfavouritekk/vdetlib.

1. Introduction

Deep learning has been widely applied to various computer vision tasks such as image classification [17, 29, 30], object detection [10, 9, 28, 30, 23, 24], semantic segmentation [21, 16], human pose estimation [33, 32, 37, 4], *etc*. Over the past few years, the performance of object detection in ImageNet and PASCAL VOC has been increased by a significant margin with the success of deep Convolutional Neural Networks (CNN). State-of-the-art methods for object detection train CNNs to classify region proposals into background or one of the object classes. However, these methods focus on detecting objects in still images. The lately introduced ImageNet challenge on object detection from video brings up a new question on how to solve the object detection problem for videos effectively and robustly. At each frame of a video, the algorithm is required to annotate bounding boxes and confidence scores on objects of each class. Although there have been methods on detecting objects in videos, they mainly focus on detecting one specific class of objects, such as pedestrians [31], cars [19], or humans with actions [13, 11]. The ImageNet challenge defines a new problem on detecting general objects in videos, which is worth studying. Similar to object detection in still images being able to assist tasks including image classification [29], object segmentation [5], and image captioning [8], accurately detecting objects in videos could possibly boost the performance of video classification, video captioning and related surveillance applications. By locating objects in the videos, the semantic meaning of a video could also be described more clearly, which results in more robust performance for video-based tasks.

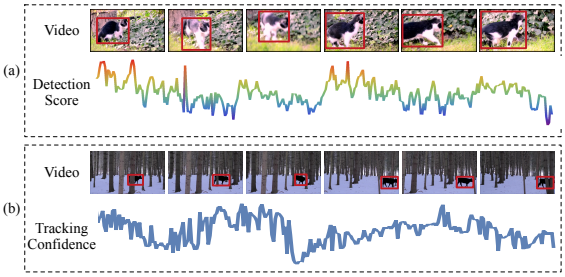

Figure 1. Challenges in object detection from video. Red boxes are ground truth annotations. (a) Still-image object detection methods have large temporal fluctuations across frames even on ground truth bounding boxes. The fluctuations may result from motion blur, video defocus, part occlusion and bad pose. Information of boxes of the same object on adjacent frames need to be utilized for object detection in video. (b) Tracking is able to relate boxes of the same object. However, due to occlusions, appearance changes and pose variations, the tracked boxes may drift to non-target objects. Object detectors should be incorporated into tracking algorithm to constantly start new tracks when drifting occurs.

Existing methods for general object detection cannot be applied to solve this problem effectively. Their performance may suffer from large appearance changes of objects in videos. For instance in Figure 1 (a), if a cat faces the camera at first and then turns back. Its back image cannot be effectively recognized as a cat because it contains little texture information and is not likely to be included in training

samples. The correct recognition result needs to be inferred from information in previous and future frames, because the appearance of an object in video is highly correlated. Since an object's location might change in the video, the location correspondences across the video should be recovered such that the correlated image patches could be well aligned into trajectories for extracting the temporal information. Besides, temporal consistency of recognition results should be regularized (Figure 1 (a)). The detection scores of a bounding-box tubelet representing an object should not change dramatically across the video.

Such requirements motivate us to incorporate object tracking into our detection framework. Deep CNNs have shown impressive performance on object tracking [36, 22], which outperform previous methods by a large margin. The large number of tracking-by-detection methods [1, 2, 3, 26] for multi-pedestrian tracking have shown that temporal information could be utilized to regularize the detection results. However, directly utilizing object tracking cannot effectively solve the VID problem either (Figure 1 (b)). In our experiments, we have noticed that directly using still-image object detectors on object tracking results has only 37.4% mean average precision (mean AP) compared to 45.3% on object proposals. The performance difference results from detectors' sensitivity to location changes and the box mismatch between tracks and object proposals. To solve this problem, we proposed a tubelet box perturbation and max-pooling process to increase the performance from 37.4% to 45.2%, which is comparable to the performance of image object proposal with only $1/38$ the number of boxes.

In this work, we propose a multi-stage framework based on deep CNN detection and tracking for object detection in videos. The framework consists of two main modules: 1) a tubelet proposal module that combines object detection and object tracking for tubelet object proposal; 2) a tubelet classification and re-scoring module that performs spatial max-pooling for robust box scoring and temporal convolution for incorporating temporal consistency. Object detection and tracking work closely in our framework. On one hand, object detection produces high-confidence anchors to initiate tracking and reduces tracking failure by spatial max-pooling. On the other hand, tracking also generates new proposals for object detection and the tracked boxes act as anchors to aggregate existing detections.

The contribution of this paper is three-fold. 1) A complete multi-stage framework is proposed for object detection in videos. 2) The relation between still-image object detection and object tracking, and their influences on object detection from video are studied in details. 3) A special temporal convolutional neural network is proposed to incorporate temporal information into object detection from video.

2. Related Works

State-of-the-art methods for detecting objects of general classes are mainly based on deep CNNs. Girshick *et al*. [10] proposed a multi-stage pipeline called Regions with Convolutional Neural Networks (R-CNN) for training deep CNN to classify region proposals for object detection. It decomposes the detection problem into several stages including bounding-box proposal, CNN pre-training, CNN fine-tuning, SVM training, and bounding box regression. Such framework has shown good performance and was adopted by other methods. Szegedy *et al*. [30] proposed the GoogLeNet with a 22-layer structure and "inception" modules to replace the CNN in the R-CNN, which won the ILSVRC 2014 object detection task. Ouyang *et al*. [23] proposed a deformation constrained pooling layer and a box pre-training strategy, which achieves an accuracy of 50.3% on the ILSVRC 2014 test set. To accelerate the training of the R-CNN pipeline, Fast R-CNN [9] is proposed, where each image patch is no longer wrapped to a fixed size before being fed into the CNN. Instead, the corresponding features are cropped from the output feature map of the last convolutional layer. In the Faster R-CNN pipeline [28], the bounding box proposals were generated by a Region Proposal Network (RPN), and the overall framework can thus be trained in an end-to-end manner. However, these pipelines are for object detection in still images. When these methods are applied to videos, they might miss some positive samples because the objects might not be of their best poses in each frame of the videos.

Object localization and co-localization [27, 25, 14, 18], which have mainly focused on the YouTube Object Dataset (YTO) [27], seems to be a similar topic to the VID task. However, there are crucial differences between the two problems. **1) Goal:** The (co)locolization problem assumes that each video contains only *one* known (weakly supervised setting) or unknown (unsupervised setting) class and only requires localizing *one* of the objects in each test frame. In VID, however, each video frame contains unknown numbers of objects instances and classes. The VID task is closer to real-world applications. **2) Metrics:** Localization metric (CorLoc [7]) is usually used for evaluation in (co)locolization, while mean average precision (mean AP) is used for evaluation on the VID task. The mean AP is more challenging to evaluate overall performances on different classes and thresholds. With these differences, the VID task is more difficult and closer to real-world scenarios. The previous works on object (co)localization in videos cannot be directly applied to VID.

There have also been methods on action localization. At each frame of human action video, the system is required to annotate a bounding box for the human action of interest. The methods that are based on action proposals are related to our work. Yu and Yuang *et al*. [38] proposed to

generate action proposals by calculating actionness scores and solving a maximum set coverage problem. Jain *et al*. [13] adopted the Selective Search strategy on super-voxels to generate tubulet proposals and proposed new features to differentiate human actions from background movements. In [11], candidate regions are fed into two CNNs to learn feature representations, which is followed by a SVM to make prediction on actions using appearance and motion cues. The regions are then linked across frames based on the action predictions and their spatial overlap.

Object tracking has been studied for decades [26, 20, 12]. Recently, deep CNNs have been used for object tracking and achieved impressive tracking accuracy [36, 22, 35]. Wang *et al*. [36] proposed to create an object-specific tracker by online selecting the most influential features from an ImageNet pre-trained CNN, which outperforms state-of-the-art trackers by a large margin. Nam *et al*. [22] trained a multi-domain CNN for learning generic representations for tracking objects. When tracking a new target, a new network is created by combining the shared layers in the pre-trained CNN with a new binary classification layer, which is online updated. However, even for the CNN-based trackers, they might still drift in long-term tracking because they mostly utilize the object appearance information within the video without semantic understanding on its class.

3. Method

In this section, we will introduce the task setting for object detection from video and give a detailed description of our framework design. The general framework of video object detection system is shown in Figure 2. The framework has two main modules: 1) a spatio-temporal tubelet proposal module and 2) a tubelet classification and re-scoring module. The two major components will be elaborated in Section 3.2 and Section 3.3.

3.1. Task setting

The ImageNet object detection from video (VID) task is similar to image object detection task (DET) in still images. There are 30 classes, which is a subset of 200 classes of the DET task. All classes are fully labeled for each video clip. For each video clip, algorithms need to produce a set of annotations (f_i, c_i, s_i, b_i) of frame number f_i, class label c_i, confidence scores s_i and bounding boxes b_i. The evaluation protocol for the VID task is the same as DET task. Therefore, we use the conventional mean average precision (mean AP) on all classes as the evaluation metric.

3.2. Spatio-temporal tubelet proposal

Objects in videos show temporal and spatial consistency. The same object in adjacent frames has similar appearances and locations. Using either existing object detection methods or object tracking methods alone cannot effectively solve the VID problem. On one hand, a straightforward application of image object detectors is to treat videos as a collection of images and detect objects on each image individually. This strategy focuses only on appearance similarities and ignores the temporal consistency. Thus the detection scores on consecutive frames usually have large fluctuations (Figure 1 (a)). On the other hand, generic object tracking methods track objects from a starting frame and usually online update detectors using samples from currently tracked bounding boxes. The detectors in tracking methods mainly focus on samples within the video and usually tends to drift due to large object appearance changes (Figure 1 (b)).

The spatio-temporal tubelet proposal module in our framework combines the still-image object detection and generic object tracking together. It has the discriminative ability from object detectors and the temporal consistency from object trackers. The tubelet proposal module has 3 major steps: 1) image object proposal, 2) object proposal scoring and 3) high-confidence object tracking.

Step 1. Image object proposal. The general object proposals are generated by the Selective Search (SS) algorithm[34]. The SS method outputs around $2,000$ object proposals on each video frame. The majority object proposals are negative samples and may not contain objects. We use the ImageNet pre-trained AlexNet [17] model provided by R-CNN to remove easy negative object proposals where all detection scores of ImageNet detection 200 classes are below a certain threshold. In our experiments, we use -1.1 as threshold and around 6.1% of the region proposals are kept, while the recall at this threshold is 80.49%. The image object proposal process is illustrated in Figure 2 (a).

Step 2. Object proposal scoring. Since the VID 30 classes are a subset of DET 200 classes, the detection models trained for the DET task can be used directly for VID classes. Our detector is a GoogLeNet [30] pre-trained on ImageNet image classification data, and fine-tuned for the DET task. Similar to the R-CNN, for each DET class, an SVM is trained with hard negative mining using the "pool5" features extracted from the model. The 30 SVM models corresponding to VID classes are used here to classify object proposals into background or one of the object classes. The higher the SVM score, the higher the confidence that the box contains an object of that class (Figure 2 (b)).

Step 3. High-confidence proposal tracking. For each object class, we track high-confidence detection proposals bidirectionally in the video clip. The tracker we choose for this task is from [36], which in our experiments shows more robust performance to object pose and scale changes. The starting detections of tracking are called "anchors", which are chosen from the most confident box proposals from Step 2. Starting from an anchor, we track backward to the first frame, and track forward to the last frame. Two tracklets

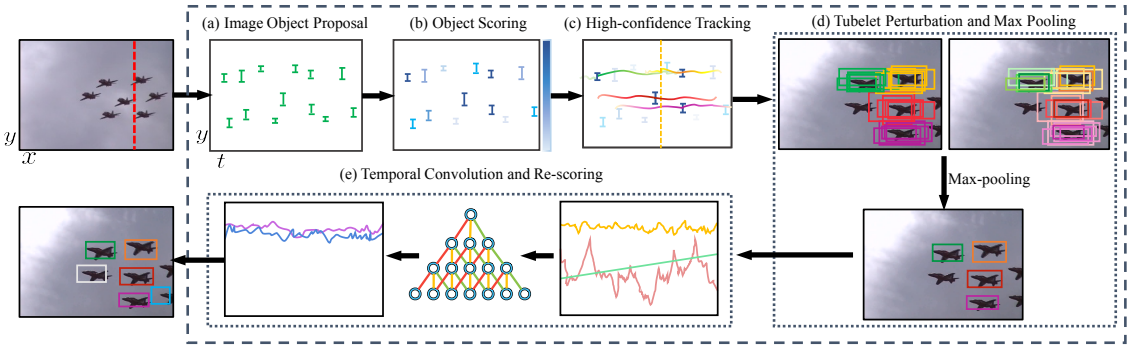

Figure 2. Video object detection framework. The proposed video object detection framework contains two major components. 1) The tubelet proposal component: (a), (b) and (c). 2) The tubelet classification and re-scoring component: (d) and (e). (a) In object proposal, class-independent proposals are generated on every frame. (b) An object scoring model outputs a detection score on every proposal. The darker the color, the higher the score. (c) High-confidence proposals are chosen as anchors for bidirectional tracking. (d) Tubelet boxes are perturbed by sampling boxes around them or replacing with original proposals. The spatial max-pooling on keeps the boxes with the highest detection scores for each tubelet box. (e) The time series of detection scores (Red), tracking score (Yellow) and anchor offsets (Green) are the inputs of the proposed temporal convolutional network. The purple line is the output of our network and blue line is the ground truth overlap value (supervision).

are then concatenated to produce a complete track. As the tracking moves away from the anchors, the tracker may drift to background and other objects, or may not keep up with the scale and pose changes of the target object. Therefore, we early stop the tracking when the tracking confidence is below a threshold (probability of 0.1 in our experiments) to reduce false positive tracklets. After getting a track, a new anchor is selected from the rest detections. Usually, high-confidence detections tend to cluster both spatially and temporally, therefore directly tracking the next most confident detection tends to result in tracklets with large mutual overlaps on the same object. To reduce the redundancy and cover as many objects as possible, we perform a suppression process similar to NMS. Detections from Step 2 that have overlaps with the existing tracks beyond a certain threshold (IOU 0.3 in our experiment) will not be chosen as new anchors. The tracking-suppression process performs iteratively until confidence values of all remaining detections are lower than a threshold (SVM score below 0 in our setting). For each video clip, such tracking process is performed for each of the 30 VID classes.

With the above three major steps, we can obtain tracks starting from high-confidence anchors for each classes. The produced tracks are tubelet proposals for tubelet classification of later part of our framework.

3.3. Tubelet classification and rescoring

After the tubelet proposal module, for each class, we have tubelets with high-confidence anchor detections. A naive approach is to classify each bounding box on the tubelets using the same method as Step 2 before. In our experiment, this baseline approach has only modest performance compared to direct still-image object detection R-CNN. The reason for that is 4-fold.

1) The overall number of bounding box proposals from tubelets is significantly smaller than those from Selective Search, which might miss some objects and result in lower recall on the test set.

2) The detector trained for object detection in still images is usually sensitive to small location changes (Figure 2 (d)) and a tracked boxes may not have a reasonable detection score even if it has large overlap with the object.

3) In the tracking process, we performed proposal suppression to reduce redundant tubelets. The tubelets are therefore more sparse compared than image proposals. This suppression may be conflict with conventional NMS. Because in conventional NMS, even a positive box has very low confidences, as long as other boxes with large mutual overlaps have higher confidence, it will be suppressed during NMS and will not affect the overall average precision. The consequence of early suppression is that some low-confidence positive boxes do not have overlaps with high confidence detections, thus are not suppressed and become false negatives.

4) The detection score along the tubelet has large variations even on ground truth tubelets (Figure 1 (a)). The temporal information should be incorporated to obtain consistent detection scores.

To handle these problems in tubelet classification, we designed the following steps to augment proposals, increase spatial detection robustness and incorporate temporal consistency into the detection scores.

Step 4. Tubelet box perturbation and max-pooling. The tubelet box perturbation and max-pooling process is to replace tubelet boxs with boxes of higher confidence. There are two kinds of perturbations in our framework.

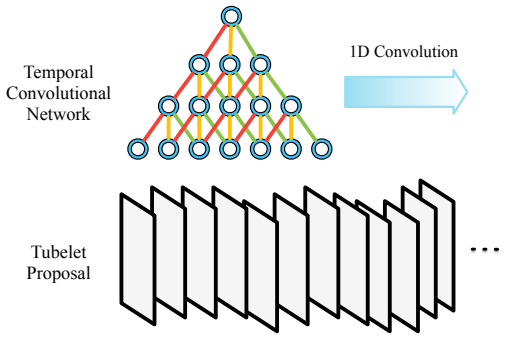

Figure 3. Temporal convolutional network (TCN). The TCN is a 1-D convolutional network that operates on tubelet proposals. The inputs are time series including detection scores, tracking scores and anchor offsets. The output values are probablities that whether each tubelet box has overlap with ground truth above 0.5.

The first method is to generate new boxes around each tubelet box on each frame by randomly perturbing the boundaries of the tubelet box. That is, we randomly sample coordinates for upper-left and bottom-right corners of a tubelet box. The random offsets for the corners are generated from two uniform distributions:

$$\Delta x \sim U(-r \cdot w, r \cdot w), \quad (1)$$

$$\Delta y \sim U(-r \cdot h, r \cdot h), \quad (2)$$

where U is uniform distribution, w and h are width and height of the tubelet box, and r is the sampling ratio hyperparameter. Higher sampling ratio means less confidence on the original box, while lower sampling ratio means more confidence on the tubelet box. We evaluated performances of different sampling configurations (Section 4).

The second perturbation method is to replace each tubelet box with original object detections that have overlaps with the tubelet box beyond a threshold. This process is to simulate the conventional NMS process. If the tubelet box is positive box with a low detection score, this process can help bring back some positive boxes to suppress this box. The higher the overlap threshold, the more confidence on the tubelet box. We find this method really effective and different overlap threshold are evaluated in Section 4.

After the box perturbation step, all augmented boxes and the original tubelet boxes are scored using the same detector in Step 2. For each tubelet box, only the augmented box with the maximum detection score is kept and used to replace the original tubelet box. The max-pooling process is to increase the spatial robustness of detector and utlize the original object detections around the tubelets.

Step 5. Temporal convolution and re-scoring. Even with the spatial max-pooling process, the detection scores along the same track might still have large variations. This naturally results in performance loss. For example, if tubelet boxes on ajacent frames all have high detection scores, it is very likely that the tublet box on this frame also has high confidence on the same object. The still-image object detection framework does not take temporal consistency into consideration.

In our framework, we propose a Temporal Convolutional Network (TCN) that uses 1-D serial features including detection scores, tracking scores, anchor offsets and generates temporally dense prediction on every tubelet box.

The structure of the proposed TCN is shown in Figure 3. It is a 4-layer 1-D fully convolution network that outputs temporally dense prediction scores on every tubelet box. For each class, we train a class-specific TCN using the tubelet features as input. The inputs are time series including detection scores, tracking scores and anchor offsets. The output values are probablities whether each tubelet box contains objects of the class. The supervision labels are 1 if the overlap with ground truth is above 0.5, and 0 otherwise.

The temporal convolution learns to generate classification prediction based on the temporal features within the receptive field. The dense 1-D labels provide richer supervision than single tubelet-level labels. During testing, the continuous classification score instead of the binary decision values. We found that this re-scoring process has consistent improvement on tubelet detection results.

4. Experiments

4.1. Dataset

ImageNet VID. We utilize the ImageNet object detection from video (VID) task dataset to evaluate the overall pipeline and individual component of the proposed framework. The framework is evaluated on the initial release of VID dataset, which consists of three distinct splits. 1) The training set contains 1952 fully-labeled video snippets ranging from 6 frames to 5213 frames per snippet. 2) The validation set contains 281 fully-labeled video snippets ranging from 11 frames to 2898 frame per snippet. 3) The test set contains 458 snippets and the ground truth annotation are not publicly available yet.

We report all results on the validation set as a common convention for object detection task.

YTO dataset. In addition to the ImageNet VID dataset, we also evaluate our proposed framework on the YTO dataset for the object localization task. The YTO dataset contains 10 object classes, which are a subset of the ImageNet VID dataset. Different from the VID dataset which contains full annotations on all video frames, the YTO dataset is weakly annotated, *i.e.* each video is only ensured to contain one object of the corresponding class, and only very few frames are annotated for each video. The weak annotation makes it infeasible to train the our models on the YTO dataset. However, since the YTO classes are a subset of the VID dataset classes, we can directly apply the trained models on

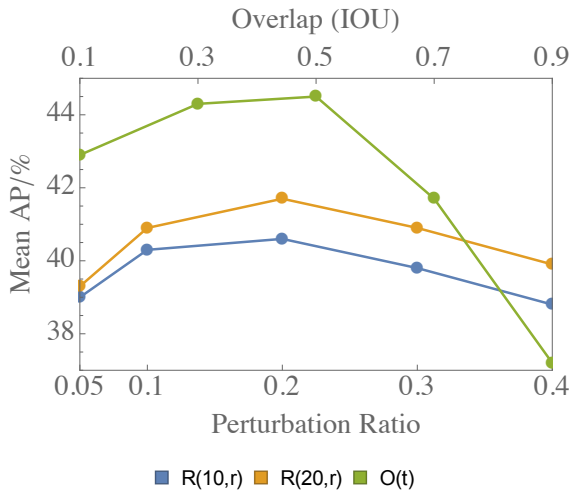

Figure 4. Performances of different max-pooling schemes. The blue and yellow lines are random sampling 10 and 20 samples per box with different perturbation ratios (bottom). The green line shows the performances of different overlap thresholds (top) for adding orginal proposals. Best viewed in color.

mean AP/%		**Sampling ratio**				
		0.05	0.1	0.2	0.3	0.4
#samples	Baseline	37.4				
	10	39.0	40.3	40.6	39.8	38.8
	20	39.3	40.9	**41.7**	40.9	39.9

Table 1. Comparison of performances of different perturbation schemes.

the YTO dataset for evaluation.

4.2. Parameter settings

Image object proposal. For image object proposal, we used the "fast" mode in Selective Search [34], and resized all input images to width of 500 pixels and mapped the generated box proposals back to original image coordinates.

The R-CNN provided AlexNet model is used to remove easy negative box proposals. We used threshold of -1.1 and remove boxes whose detections scores of all DET 200 are below the threshold. This process kept 6.1% of all the proposals (about 96 boxes per image).

Tracking. The early stopping tracking confidence is set to probability 0.1. Therefore, if the tracking confidence is below 0.1, the tracklet is terminated. The minimum detection score for a new tracking anchor is 0. If no more detection beyond this threshold, the whole tracking process ends for this class. The track-detection suppression overlap is set to 0.3. For each video snippet, we chose at most 20 anchors for tracking for each class.

Tubelet perturbation. For tubelet box perturbation, we denote $R(n, r)$ for random perturbation with perturbation ratio r and n samples per tubelet box, and $O(t)$ for adding original proposals whose overlaps with the tubelet boxes beyond threshold t.

Overlap (IOU)	0.1	0.3	0.5	0.7	0.9	Baseline
mean AP/%	42.9	44.3	**44.5**	41.7	37.2	37.4

Table 2. Comparison of performances of different IOU threshold for adding original detections.

Layer	Input	conv1	conv2	conv3	conv4	softmax
Kernel size $(n \times t)$		256×5	256×5	256×7	2×3	
Output size $(c \times t)$	3×50	256×50	256×50	256×50	2×50	2×50

Table 3. Temporal convolutional network (TCN) structure.

Different combinations of perturbation ratios and sampling numbers are evaluated as shown in Table 1 and Figure 4. $R(20, 0.1)$ and $R(20, 0.2)$ are chosen for later components. For $O(t)$ schemes, $O(0.1)$ to $O(0.9)$ are evaluated (Figure 4 and Table 2). $O(0.5)$ is chosen for the framework.

Temporal convolutional network The TCN in our experiments has 4 convolutional layers, the network structure is shown in Table 3. The network initialization parameter and optimization parameter such as learning rate are manually adjusted on one class and remained unchanged for all 30 classes.

The network raw detection score, tracking score and absolute anchor offsets (which is normalized by length of the track) are used as input feature for the TCN, without other preprocessing steps.

5. Results

5.1. Quantitative Results on VID

Tubelet proposal and baseline performance. After obtaining the tubelet proposals, a straight-forward baseline approach for tubelet scoring is to directly evaluate tubelet boxes with the object detector for still images. The performance of this approach is 37.4% in mean AP. In comparison, the still-image object detection result is 45.3% in mean AP.

Tubelet perturbation and spatial max-pooling. The performances of different tubelect box perturbation and max-pooling schemes are shown in Table 1 and Table 2. From the tables, we can see that in most settings, both the random sampling and adding original proposals improves over the baseline approach. Also, if the perturbation is too large or too small, the performance gain is small. The reasons are: 1) large perturbation may result in replacing the tubelet box with a box too far away from it, and on the other hand, 2) small perturbation may obtain redundant boxes that reduce the effect of spatial max-pooling.

The best performance of random sampling scheme is 41.7% of $R(20, 0.2)$. For replacing with original propos-

Method	airplane	antelope	bear	bicycle	bird	bus	car	cattle	dog	domestic cat	elephant	fox	giant panda	hamster	horse	lion
Still image	64.50	71.40	42.60	36.40	18.80	62.40	**37.30**	47.60	15.60	49.50	**66.90**	66.30	58.20	74.10	25.50	29.00
Baseline	64.10	61.80	37.60	37.80	19.90	51.10	28.10	46.50	9.50	44.10	52.30	56.40	52.00	56.40	21.20	29.70
s1: R(20,0.1)	68.10	66.60	39.80	35.30	20.00	53.80	30.50	47.20	12.80	48.20	56.40	62.50	53.40	60.80	26.20	30.70
s2: O(0.5)	70.80	71.60	**43.20**	37.00	20.40	**64.40**	33.80	48.00	16.70	**54.50**	62.70	70.80	57.10	73.30	26.00	30.70
s3: s1 + s2	71.30	70.90	42.50	37.80	21.00	61.70	35.00	48.10	16.60	53.20	63.00	70.00	58.00	70.90	27.00	33.50
TCN: s3	**72.70**	**75.50**	42.20	**39.50**	**25.00**	64.10	36.30	**51.10**	**24.40**	48.60	65.60	**73.90**	**61.70**	**82.40**	**30.80**	**34.40**

Method	lizard	monkey	motorcycle	rabbit	red panda	sheep	snake	squirrel	tiger	train	turtle	watercraft	whale	zebra	mean AP	#win
Still image	**68.70**	1.90	50.80	34.20	**29.40**	**59.00**	**43.70**	1.80	33.00	**56.60**	66.10	61.10	24.10	64.20	45.30	7
Baseline	38.70	1.90	41.40	34.10	19.60	45.50	10.90	1.30	12.80	48.10	64.90	52.40	17.20	63.60	37.40	0
s1: R(20,0.1)	51.80	1.50	44.60	33.30	15.10	51.40	26.20	1.80	18.70	49.00	65.80	57.10	22.20	67.40	40.60	0
s2: O(0.5)	62.80	**2.10**	52.60	33.50	9.90	58.50	26.00	2.30	30.70	54.70	**67.70**	62.50	23.60	66.30	44.50	5
s3: s1 + s2	64.30	1.90	51.40	34.10	15.50	57.90	40.20	2.40	31.60	52.20	67.50	63.10	24.50	67.10	45.10	0
TCN: s3	54.20	1.60	**61.00**	**36.60**	19.70	55.00	38.90	**2.60**	**42.80**	54.60	66.10	**69.20**	**26.50**	**68.60**	**47.50**	**18**

Table 4. Performances of different methods and experimental settings. "s#" stands for different settings. $R(n, r)$ represents random sampling pertubation scheme with perturbation ration of r and n samples per tubelet box. $O(t)$ represents adding original proposals with overlap larger than threshold r.

als, the best result is 44.5% of $O(0.5)$. It is worth noticing that the tubelet perturbation and max-pooling scheme does not increase the overall boxes of tubelet proposals but replaces original tubelet boxes with nearby ones with the highest confidences.

We also investigated the complementary properties of the two perturbation schemes. The perturbed tubelets from the best settings of the both schemes (41.7% model from $R(20, 0.2)$ and 44.5% model from $O(0.5)$) are combined for evaluation. The direct combination doubles the number of tubelets, and the performance increases from 41.7% and 44.5% to 45.2%, which is comparable to still-image object detection result with much fewer proposals on each image (around $1:38$).

Temporal convolution. Using the tubelets proposals, we trained a TCN for each of the 30 classes for re-scoring. We use the continuous values of Sigmoid foreground scores as the confidence values for the tubelet boxes in evaluation.

For the baseline 37.4% model, the performance increases to 39.4% by 2%. On the best single perturbation scheme proposal ($O(0.5)$), the performance increases from 44.5% to 46.4%. For combined tubelet proposals from two perturbation schemes, a 45.2% model with $R(20, 0.2)$ and $O(0.5)$ increases the performance to 47.4, while a 45.1 model with $R(20, 0.1)$ and $O(0.5)$ increases to 47.5%.

From the results we can see that our temporal convolution network using detection scores, tracking scores and anchor offsets provides consistent performance improvement (around 2 percents in mean AP) on the tubelet proposals.

Overall, the best performance on tubelet proposals by our proposed method is 47.5%, 2.2 percents increase from still-image object detection framework with only $1/38$ the number of boxes for evaluation.

5.2. Qualitative Results on VID

Tubelet proposal. The tubelet proposal results are shown in Figure 5. Figure 5 (a) shows the positive tubelets obtained from tubelet proposal module, and Figure 5 (b) shows the negative samples.

From the figure we can see, positive samples usually aggregate around objects while still appear sparse compared to dense proposals from Selective Search. The sparsity comes from the track-proposal suppression process performed in tracking to ensure the tubelet covers as many objects as possible.

With the frame index increases, some tracks will disappear while others may be added, which results from the early stopping for low tracking confidence and new anchor selections.

As for the negative samples, the tubelet are much fewer (in fact, some videos do not have tubelet proposals for some classes) and isolated. This is because we only start tracking on high-confident anchors. This largely reduces the number of false positives and significantly save inference time in later steps in the framework.

Temporal convolution. In Figure 6, we show some examples of results of temporal convolution. Each plot shows the tubelet scores (detection score, tracking score and anchor offsets) and the output probability scores of the TCN network.

The detection score shows significant variations across frames. A tubelet box may have significantly low detection score even if adjacent frames have high detection values. However, after temporal convolution, the output probability curve are much more temporally consistent. Compare to detection scores, the probability output of our network conforms better with the ground truth overlaps, which shows

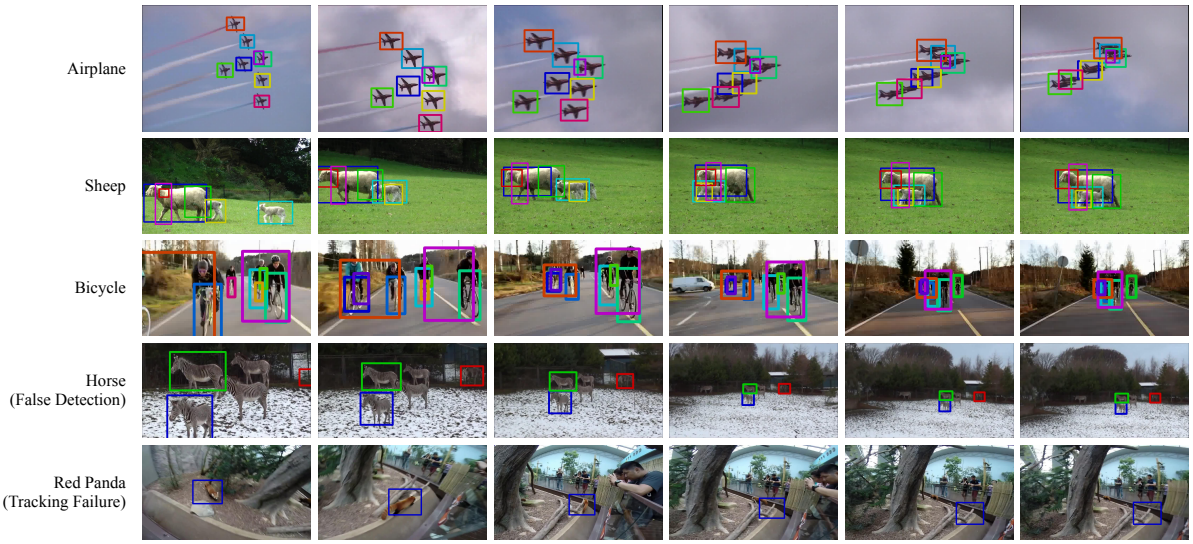

Figure 5. Qualitative results of tubelet proposals. The first three rows are positive bounding boxes of tubelet proposals generated by our framework. The proposal boxes usually aggregate on ground truth objects while keep sparsely allocated to cover as many objects as possible. The last two rows shows some failure cases of tubelet proposals. The first kind of failure cases are false detections (for example, mis-detect zebras as horses). The second kind of failure cases are tracking failures. Tracker drifts to background objects due to large scale changes of target objects while the mis-tracked targets are not confident enough to start new tracking processes.

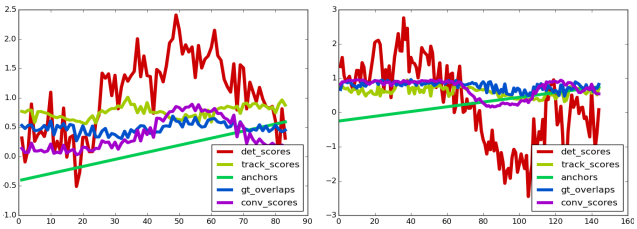

Figure 6. Qualitative results of temporal convolution. The time series of detection scores, tracking scores and absolute values of anchor offsets are the inputs for TCN. The blue line are overlaps with ground truth annotations and purple lines are the output of TCN. The detection scores have large temporal variations while the TCN output has temporal consistency and comply better to ground truth overlaps.

the effectiveness of our re-scoring module.

5.3. Evaluation on YouTube-Objects (YTO) dataset

In order to show the effectiveness of our proposed framework, we applied the models trained on the VID task directly on the YTO dataset and compared with the state-of-the-art works in Table 5. The localization metric CorLoc [7] is used for evaluation as a convention on YTO.

Table 5. Localization performances on the YTO dataset

Method	aero	bird	boat	car	cat	cow	dog	horse	mbike	train	Avg.
Prest *et al.* [27]	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5
Joulin *et al.* [14]	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0
Kwak *et al.* [18]	56.5	66.4	58.0	76.8	39.9	**69.3**	50.4	56.3	53.0	31.0	55.7
Baseline	92.4	68.4	85.4	75.8	**77.3**	18.6	87.2	87.3	84.2	72.8	74.9
Ours (TCN:s3)	**94.1**	**69.7**	**88.2**	**79.3**	76.6	18.6	**89.6**	**89.0**	**87.3**	**75.3**	**76.8**

From the table, we can see that our proposed framework outperforms by a large margin. This is because the ImageNet datasets (CLS, DET and VID) provide rich supervision for feature learning, and the trained networks have good generalization capability on other datasets.

The full framework has around 2% improvement over the baseline approach on the YTO dataset, which is consistent with the results on VID.

6. Conclusion

In this work, we propose a complete multi-stage pipeline for object detection in videos. The framework efficiently combines still-image object detection with generic object tracking for tubelet proposal. Their relationship and contributions are extensively investigated and evaluated. Based on tubelet proposals, different perturbation and scoring schemes are evaluated and analyzed. A novel temporal convolutional network is proposed to incorporate temporal consistency and shows consistent performance improvement over still-image detections. Based on this work, a more advanced tubelet-based framework is further developed which won the ILSVRC2015 ImageNet VID challenge with provided data [15].

Acknowledgement

This work is partially supported by SenseTime Group Limited, and the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project Nos. CUHK14206114, CUHK14205615, CUHK14203015, CUHK14207814).

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. *CVPR*, 2008. 2

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. *CVPR*, 2012. 2

[3] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. *CVPR*, 2014. 2

[4] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *CVPR*, 2016. 1

[5] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *ICCV*, 2015. 1

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009. 1

[7] T. Deselaers, B. Alexe, and V. Ferrari. Localizing Objects While Learning Their Appearance. *ECCV*, 2010. 2, 8

[8] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. *CVPR*, 2014. 1

[9] R. Girshick. Fast r-cnn. *ICCV*, 2015. 1, 2

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014. 1, 2

[11] G. Gkioxari and J. Malik. Finding action tubes. *CVPR*, 2014. 1, 3

[12] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. *CVPR*, 2015. 3

[13] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. *CVPR*, 2014. 1, 3

[14] A. Joulin, K. Tang, and L. Fei-Fei. Efficient Image and Video Co-localization with Frank-Wolfe Algorithm. *ECCV*, 2014. 2, 8

[15] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, and W. Ouyang. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *arXiv preprint*, 2016. 8

[16] K. Kang and X. Wang. Fully convolutional neural networks for crowd segmentation. *arXiv preprint arXiv:1411.4464*, 2014. 1

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, pages 1097–1105, 2012. 1, 3

[18] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised Object Discovery and Tracking in Video Collections. *ICCV*, 2015. 2, 8

[19] B. Li, T. Wu, and S.-C. Zhu. Integrating context and occlusion for car detection by hierarchical and-or model. *ECCV*, 2014. 1

[20] Y. Li, J. Zhu, and S. C. Hoi. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. *CVPR*, 2015. 3

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[22] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. *arXiv:1510.07945*, 2015. 2, 3

[23] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, et al. DeepID-net: Deformable deep convolutional neural networks for object detection. *CVPR*, 2015. 1, 2

[24] W. Ouyang, X. Wang, C. Zhang, and X. Yang. Factors in finetuning deep model for object detection with long-tail. In *CVPR*, 2016. 1

[25] A. Papazoglou and V. Ferrari. Fast Object Segmentation in Unconstrained Video. *ICCV*, 2013. 2

[26] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. *CVPR*, 2014. 2, 3

[27] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. *CVPR*, 2012. 2, 8

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 2015. 1, 2

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014. 1

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015. 1, 2, 3

[31] Y. Tian, P. Luo, X. Wang, and X. Tang. Pedestrian detection aided by deep learning semantic tasks. *CVPR*, 2014. 1

[32] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015. 1

[33] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 1

[34] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 3, 6

[35] L. Wang, W. Ouyang, X. Wang, and L. Huchuan. Stct: Sequentially training convolutional networks for visual tracking. In *CVPR*, 2016. 3

[36] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. *ICCV*, 2015. 2, 3

[37] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, 2016. 1

[38] G. Yu and J. Yuan. Fast action proposals for human action detection and search. *CVPR*, 2015. 2