

Temporal Context Network for Activity Localization in Videos

Xiyang Dai¹ Bharat Singh¹ Guyue Zhang² Larry S. Davis¹ Yan Qiu Chen²
¹University of Maryland ²Fudan University
 College Park, MD Shanghai, China
 xdai, bharat, lsd@umiacs.umd.edu guyuezhang13, chenyyq@fudan.edu.cn

Abstract

We present a Temporal Context Network (TCN) for precise temporal localization of human activities. Similar to the Faster-RCNN architecture, proposals are placed at equal intervals in a video which span multiple temporal scales. We propose a novel representation for ranking these proposals. Since pooling features only inside a segment is not sufficient to predict activity boundaries, we construct a representation which explicitly captures context around a proposal for ranking it. For each temporal segment inside a proposal, features are uniformly sampled at a pair of scales and are input to a temporal convolutional neural network for classification. After ranking proposals, non-maximum suppression is applied and classification is performed to obtain final detections. TCN outperforms state-of-the-art methods on the ActivityNet dataset and the THU-MOS14 dataset.

1. Introduction

Recognizing actions and activities in videos is a long studied problem in computer vision [2, 10, 3]. An action is defined as a short duration movement such as jumping, throwing, kicking. In contrast, activities are more complex. An activity has a beginning, which is triggered by an action or an event, which involves multiple actions, and an end, which involves another action or an event. For example, an activity like “assembling a furniture” could start with unpacking boxes, continue by putting different parts together and end when the furniture is ready. Since videos can be arbitrarily long, they may contain multiple activities and therefore, temporal localization is needed. Detecting human activities in videos has several applications in content based video retrieval for web search engines, reducing the effort required to browse through lengthy videos, monitoring suspicious activity in video surveillance etc. While localizing objects in images is an extensively studied problem, localizing activities has received less attention. This is primarily because performing localization in videos is com-

putationally expensive [6] and well annotated large datasets [5] were unavailable until recently.

Current object detection pipelines have three major components - proposal generation, object classification and bounding box refinement [25]. In [6, 29] this pipeline was adopted for deep learning based action detection as well. LSTM is used to embed a long video into a single feature vector which is then used to score different segment proposals in the video [6]. While a LSTM is effective for capturing local context in a video [31], learning to predict the start and end positions for all activity segments using the hidden state of a LSTM is challenging. In fact, in our experiments we show that even a pre-defined set of proposals at multiple scales obtains better recall than the temporal segments predicted by a LSTM on the ActivityNet dataset.

In [29], a ranker was learned on multiple segments of a video based on overlap with ground truth segments. However, a feature representation which does not integrate information from a larger temporal scale than a proposal lacks sufficient information to predict whether a proposal is a good candidate or not. For example, in Figure 1, the red and green solid segments are two proposals which are both completely included within an activity. While the red segment is a good candidate, the green is not. So, although a single scale representation for a segment captures sufficient information for recognition, it is inadequate for detection. To capture information for predicting activity boundaries, we propose to explicitly sample features both at the scale of the proposal and also at a higher scale while ranking proposals. We experimentally demonstrate that this has significant impact on performance when ranking temporal activity proposals.

By placing proposals at equal intervals in a video which span multiple temporal scales, we construct a set of proposals which are then ranked using features sampled from a pair of scales. A temporal convolution network is applied over these features to learn background and foreground probabilities. The top ranked proposals are then input to a classification network which assigns individual class probabilities to each segment proposal.

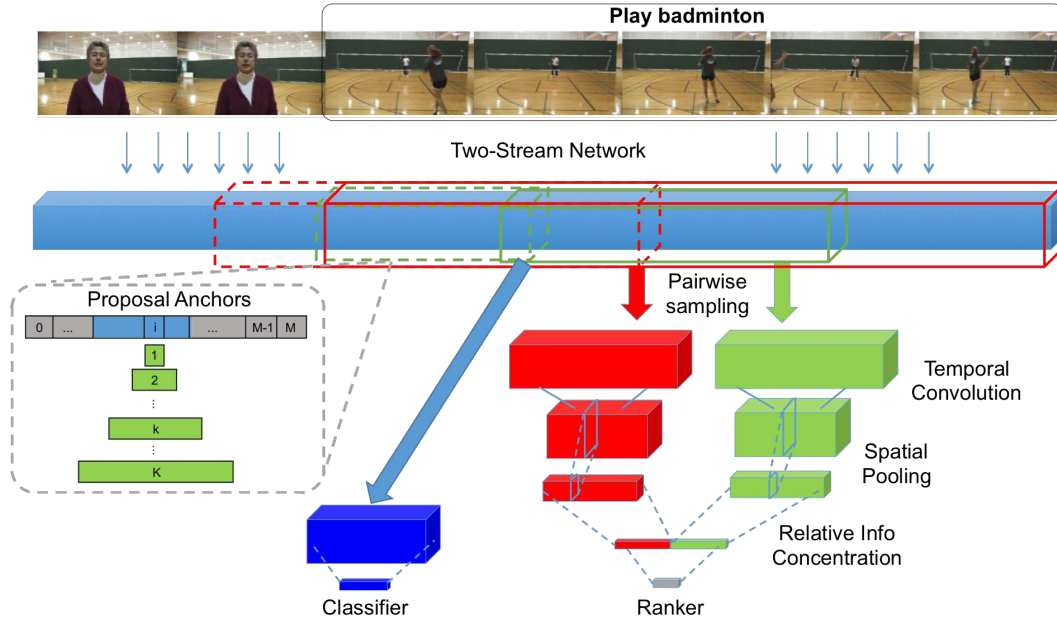


Figure 1. Given a video, a two stream network is used to extract features. A pair-wise sampling layer samples features at two different resolutions to construct the feature representation for a proposal. This pairwise sampling helps to obtain a better proposal ranking. A typical sliding window approach (Green line box) can miss the context boundary information when it lies inside the activity. However, the proposed pairwise sampling with a larger context window (Red line box) will capture such information and yield better proposal ranking. These pair-wise features are then input to a ranker which selects proposals for classification. The green boxes on the left represent K different proposals which are placed uniformly in a video.

2. Related Work

Wang and Schmidt [33] introduced Dense Trajectories (DT), which have been widely applied in various video recognition algorithms. For trimmed activity recognition, extracting dense trajectories and encoding them by using Fisher Vectors has been widely used [1, 34, 14, 12, 23, 37]. For action detection, [41] constructed a pyramid of score distribution features (PSDF) as a representation for ranking segments of a video in a dense trajectories based pipeline. However, for large datasets, these methods require significant computational resources to extract features and build the feature representation after features are extracted. Because deep learning based methods provide better accuracy with much less computation, hand-crafted features have become less popular.

For object detection in images, proposals are a critical elements for obtaining efficient and accurate detections [26, 25]. Motivated by this approach, Jain et al. [13] introduced action proposals which extends object proposals to videos. For spatio-temporal localization of actions, multiple methods use spatio-temporal region proposals [9, 22, 7, 39]. However, these methods are typically applied to datasets containing short videos, and hence the major focus has been on spatial localization rather than temporal localization. Moreover, spatio-temporal localization requires training data containing frame level bounding box annotations.

For many applications, simply labeling the action boundaries in the video is sufficient, which is a significantly less cumbersome annotation task.

Very recently, studies focusing on temporal segments which contain human actions have been introduced [18, 4, 29, 17, 31]. Similar to grouping techniques for retrieving object proposals, Heilbron et al. [4] used a sparse dictionary to encode discriminative information for a set of action classes. Mettes et al. [18] introduced a fragment hierarchy based on semantic visual similarity of contiguous frames by hierarchical clustering, which was later used to efficiently encode temporal segments in unseen videos. In [31], a multi-stream RNN was employed along with tracking to generate frame level predictions to which simple grouping was applied at multiple detection thresholds for obtaining detections.

Methods using category-independent classifiers to obtain many segments in a long video are more closely related to our approach. For example, Shou et al. [29] exploit three segment-based 3D ConvNets: a proposal network for identifying candidate clips that may contain actions, a classification network for learning a classification model and a localization network for fine-tuning the learned classification network to localize each action instance. Escorcia et al. [6] introduce Deep Action Proposals (DAPs) and use a LSTM to encode information in a fixed clip (512 frames) of

a video. After encoding information in the video clip, the LSTM scores K (64) predefined start and end positions in that clip. The start and end positions are selected based on statistics of the video dataset. We show that our method performs better than global representations like LSTMs which create a single feature representation for all scales in a video for localization of activities. In contemporary work, Shou et al. [28] proposed a convolutional-de-convolutional (CDC) network by combining temporal upsampling and spatial downsampling for activity detection. Such an architecture helps in precise localization of activity boundaries. We show that the activity proposals generated by our method can further improve CDC's performance.

Context has been widely used in various computer vision algorithms. For example, it helps in tasks like object detection [8], semantic segmentation [21], referring expressions [40] etc. In videos it has been used for action and activity recognition [11, 38]. However, for temporal localization of activities, existing methods do not employ temporal context, which we show is critical for solving this problem.

3. Approach

Given a video \mathcal{V} , consisting of T frames, TCN generates a ranked list of segments s_1, s_2, \dots, s_N , each associated with a score. Each segment s_j is a tuple t_b, t_e , where t_b and t_e denote the beginning and end of a segment. For each frame, we compute a D dimensional feature vector representation which is generated using a deep neural network. An overview of our method is shown in Figure 2.

3.1. Proposal Generation

Our goal in this step is to use a small number of proposals to obtain high recall. First, we employ a temporal sliding window of a fixed length of L frames with 50% overlap. Suppose each video \mathcal{V} has M window positions. For each window at position i ($i \in [0, M]$), its duration is specified as a tuple (b_i, e_i) , where b_i and e_i denote the beginning and end of a segment. We then, generate K proposal segments (at K different scales) at each position i . For $k \in [1, K]$, the segments are denoted by (b_i^k, e_i^k) . Also, the duration of each segment, L_k , increases as a power of two, i.e $L_{k+1} = 2L_k$. This allows us to cover all candidate activity locations that are likely to contain activities of interests, and we refer them as activity proposals, $P = \{(b_i^k, e_i^k)\}_{i=0, k=1}^{M, K}$. Figure 1 illustrates temporal proposal generation. When a proposal segment meets the boundary of a video, we use zero-padding.

3.2. Context Feature Representation

We next construct a feature representation for ranking proposals. We use all the features $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$ of the untrimmed video as a feature representation for the video. For the k^{th} proposal at window position i ($P_{i,k}$), we

uniformly sample from \mathcal{F} to obtain a D dimensional feature representation $Z_{i,k} = \{z_1, z_2, \dots, z_n\}$. Here, n is the number of features which are sampled from each segment. To capture temporal context, we again uniformly sample features from \mathcal{F} , but this time, from $P_{i,k+1}$ — the proposal at the next scale and centered at the same scale. Note that we do not perform average or max-pooling but instead sample a fixed number of frames regardless of the duration of $P_{i,k}$.

Logically, a proposal can fall into one of four categories:

- It is disjoint from a ground-truth interval and therefore, the next scale's (larger) label is irrelevant
- It includes a ground-truth interval and the next-scale has partial overlap with that ground truth interval.
- It is included in a ground-truth interval and the next level has significant overlap with the background (i.e., it is larger than the ground truth interval).
- It is included in a ground-truth interval and so is the next level.

A representation which only considers features inside a proposal would not consider the last two cases. Hence, whenever a proposal is inside an activity interval, it would not be possible to determine where the activity ends by only considering the features inside the proposal. Therefore, using a context based representation is critical for temporal localization of activities. Additionally, based on how much background the current and next scales cover, it becomes possible to determine if a proposal is a good candidate.

3.3. Sampling and Temporal Convolution

To train the proposal network, we assign labels to proposals based on their overlap with ground truth, as follows,

$$Label(S_j) = \begin{cases} 1, & iou(S_j, GT) > 0.7 \\ 0, & iou(S_j, GT) < 0.3 \end{cases} \quad (1)$$

where $iou(\cdot)$ is intersection over union overlap and GT is a ground truth interval. During training, we construct a mini batch with 1024 proposals with a positive to negative ratio of 1:1.

Given a pair of features $Z_{i,k}, Z_{i,k+1}$, from two consecutive scales, we apply temporal convolution to features sampled from each temporal scale separately to capture context information between scales, as shown in Figure 2. A temporal Convolutional Neural Network [16] enforces temporal consistency and obtains consistent performance improvements over still-image detections. To aggregate information across scales, we concatenate the two features to obtain a fixed dimensional representation. Finally, two fully connected layers are used to capture context information across scales. A two-way Softmax layer followed by cross-entropy

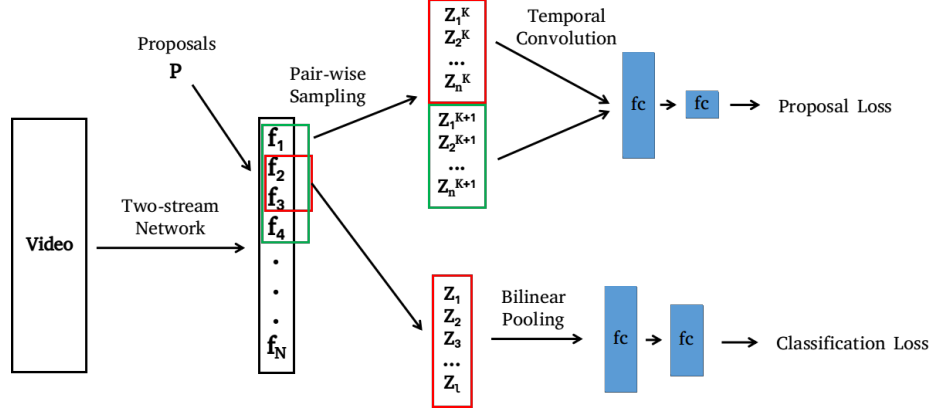


Figure 2. Temporal Context Network applies a two stream CNN on a video for obtaining an intermediate feature representation.

loss is used at the end to map the predictions to labels (proposal or not).

3.4. Classification

Given a proposal with a high score, we need to predict its action class. We use bilinear pooling by computing the outer product of each segment feature, and average pool them to obtain the bilinear matrix $bilinear(\cdot)$. Given features $\hat{Z} = [z_1, z_2, \dots, z_l]$ within a proposal, we conduct bilinear pooling as follows:

$$bilinear(\hat{Z}) = \sum_{i=1}^l \hat{Z}_i^T \hat{Z}_i \quad (2)$$

For classification, we pool all the features l which are inside the segment and do not perform any temporal sampling. We pass this vectorized bilinear feature $x = bilinear(\hat{Z})$ through a mapping function with signed square root and l^2 normalization [24]:

$$\phi(x) = \frac{sign(x)\sqrt{x}}{\|sign(x)\sqrt{x}\|_2} \quad (3)$$

We finally apply a fully connected layer and use a 201-way (200 action classes plus background) Softmax layer at the end to predict class labels. We again use the cross entropy loss function for training. During training, we sample 1024 proposals to construct a mini batch. To balance training, 64 samples are selected as background in each mini-batch. For assigning labels to video segments, we use the same function which is used for generating proposals,

$$Label(S_j) = \begin{cases} lb, & iou(S_j, GT) > 0.7 \\ 0, & iou(S_j, GT) < 0.3 \end{cases} \quad (4)$$

where $iou(\cdot)$ is intersection over union overlap, GT is ground truth and lb is the most dominant class with in proposal S_j . We use this classifier for the ActivityNet dataset but this can be replaced with other classifiers as well.

4. Experiments

In this section, we provide analysis of our proposed temporal context network. We perform experiments on the ActivityNet and THUMOS14 datasets.

4.1. Implementation details

We implement the network based on a customized Caffe repository with Python interface. All evaluation experiments are performed on a workstation with a Titan X (Maxwell) GPU. We initialize our network with pre-trained TSN models [35] and fine-tune them on both action labels and foreground/background labels to capture “actionness” and “backgroundness”. Later, we concatenate these together as high-level features input to our proposal ranker and classifier. For the proposal ranker, we use temporal convolution with a kernel size of 5 and a stride of 1, followed by ReLU activation and average pooling with size 3 and stride 1. The temporal convolution responses are then concatenated and mapped to a fully connected layer with 500 hidden units, which is used to predict the proposal score. To evaluate our method on the detection task, we generate top K proposals (K is set to 20, we apply non-maximum suppression to filter out similar proposals, using an NMS threshold set as 0.45) and classify them separately. While classifying proposals, we also fuse two global video level priors using ImageNet shuffle features [19] and “actionness” features to further improve classification performance, as shown in [32]. We also perform an ablation study for different components of classification. For training the proposal network, we use a learning rate 0.1. For the classification network, we set learning rate to 0.001. For both cases, we use a momentum of 0.9 and 5e-5 weight decay.

4.2. ActivityNet Dataset

ActivityNet [5] is a recently released dataset which contains 203 distinct action classes and a total of 849 hours of videos collected from YouTube. It consists of both trimmed

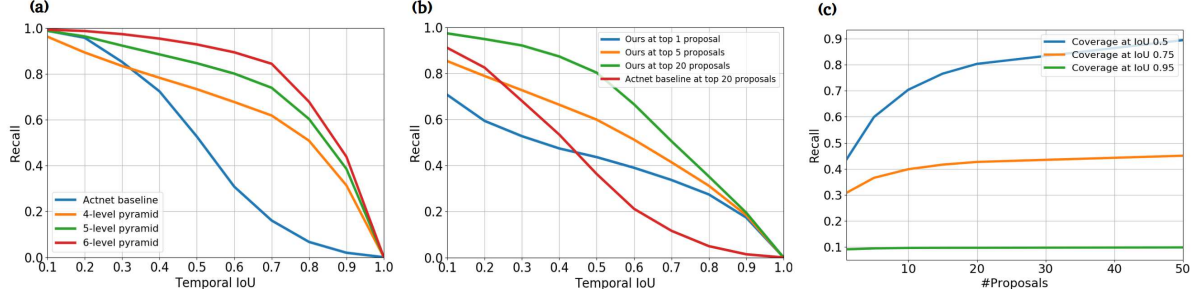


Figure 3. Performance of our proposal ranker on ActivityNet validation set. (a) The Recall vs IoU for pyramid proposal anchors; (b) The Recall vs IoU for our ranker at 1, 5, 20 proposals; (c) Recall vs number of proposals for our ranker at IoU 0.5, 0.75 and 0.95

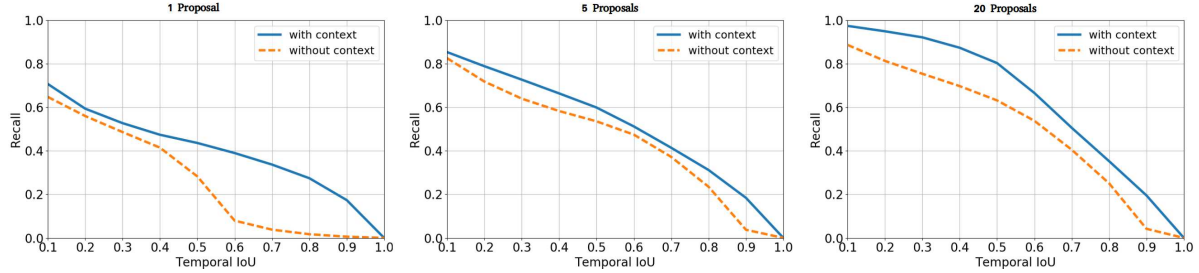


Figure 4. The effectiveness of context-based proposal ranker is shown in these plots. The Recall vs IoU plots show ranker performance at 1, 5, 20 proposals with and without context on ActivityNet validation set

and untrimmed videos. Each trimmed video contains a specific action with annotated segments. Untrimmed videos contain one or more activities with background involved. On average, each activity category has 137 untrimmed videos. Each video on average has 1.41 activities which are annotated with beginning and end points. This benchmark is designed for three applications: untrimmed video classification, trimmed activity classification, and untrimmed activity detection. Here, we evaluate our performance on the detection task in untrimmed videos. We use the mean average precision (mAP) averaged over multiple overlap thresholds to evaluate detection performance. Since test labels of ActivityNet are not released, we perform ablation studies on the validation data and test our full model on the evaluation server.

Proposal anchors We sample pair-wise proposals within a temporal pyramid. In Figure 3(a), we present the recall for the pyramid proposal anchors on ActivityNet validation set with three different levels. This figure shows the theoretical best recall one can obtain using such a pyramid. Notice that even with a 4-level pyramid with 64 proposals in total, the coverage is already better than the baseline provided in the challenge, which uses 90 proposals. This ensures our proposal ranker’s performance is high with a low number of proposals.

Performance of our ranker We evaluate our ranker with different numbers of proposals. Figure 3(b) shows the average recall at various overlap thresholds with top 1, top 5 and top 20 proposals. Even when using *one* proposal,

	mAP@.5	mAP@.75	mAP@.95
without context	15.91	3.11	0.13
with context	36.17	21.12	3.89

Table 1. Evaluation on the influence with and without context on ActivityNet validation set

our ranker outperforms the ActivityNet proposal baseline by a significant margin when the overlap threshold is greater than 0.5. With top 20 proposals, our ranker can squeeze out most of the performance from pyramid proposal anchors. We also evaluate the performance of our ranker by measuring recall as the number of proposals varies (shown in Figure 3(c)). Recall at IoU 0.5 increases to 90% with just 20 proposals. At higher IoU, increasing the number of proposals does not increase recall significantly.

Effectiveness of temporal context We contend that temporal context for ranking proposals is critical for localization. To evaluate this claim, we conduct several experiments. In Figure 4, we compare the performance of the ranker with and without temporal context. Using only the best proposal, without context, the recall drops significantly at high IoU ($\text{IoU} > 0.5$). This shows that for precise localization of boundaries, temporal context is critical. Using top 5 and top 20 proposals, without context, the recall is marginally worse. This is expected because as the number of proposals increases, there is a higher likelihood of one having a good overlap with a ground-truth. Therefore, recall results using a single proposal are most informative.

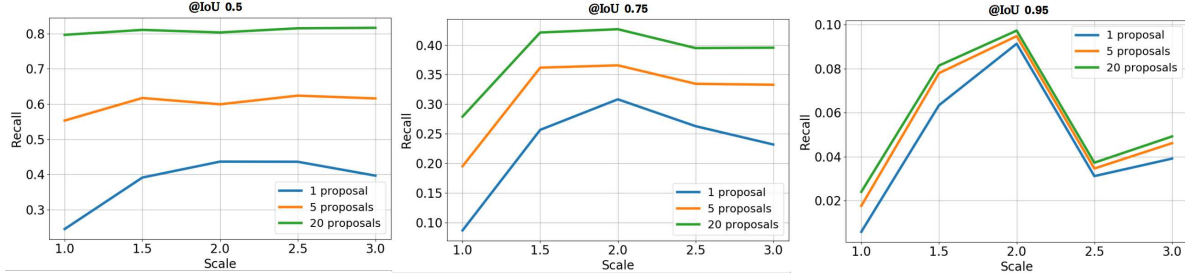


Figure 5. Comparing the ranker performance using different relative scale for context based proposals on ActivityNet validation set

Context Scale	mAP@.5	mAP@.75	mAP@.95
1	15.91	3.11	0.13
1.5	30.51	15.56	2.23
2	36.17	21.12	3.89
2.5	36.04	17.08	0.92
3	33.29	14.35	1.03

Table 2. Impact of varying temporal context at different overlap thresholds on ActivityNet validation set

We also compute detection metrics on the ActivityNet validation set to evaluate the influence of context. Table 1 also shows that detection mAP is much higher when using the ranker with context based proposals. These experiments demonstrate the effectiveness of our method.

Varying context window for ranking proposals Another important component for ranking proposals is the scale of context features which are associated with the proposal. Consider a case in which a proposal is contained within the ground truth interval. If the context scale is large, the ranker may not be able to distinguish between good and bad proposals, as it always see a significant amount of background. If the scale is small, there may be not enough context to determine if the proposal is contained within the ground truth or not. Therefore, we conduct an experimental study by varying the scale of context features while ranking proposals. In Figure 5, we observe that the performance improves up to a scale of 2. We evaluate the performance of the ranker at different scales on the ActivityNet validation set. In Table 2 we show the impact of varying temporal context at different overlap thresholds, which validates our claim that adding more temporal context would hurt performance, but not using context at all would reduce performance by a much larger margin. For example, changing the scale from 2 to 3 only drops the performance by 3% but changing it from 1.5 to 1 decreases mAP by 15% and 12% respectively.

Influence of number of proposals We also evaluate the influence of the number of proposals on detection performance. Table 3, shows that our method doesn't require a large number of proposals to improve its highest mAP. This demonstrates the advantages of both our proposal ranker

#Proposal/Video	mAP@.5	mAP@.75	mAP@.95
1	25.70	16.08	2.80
5	34.13	20.72	3.89
10	35.52	21.02	3.89
20	36.17	21.12	3.89
50	36.44	21.15	3.90

Table 3. Impact of number proposals on mAP on ActivityNet validation set

Components			mAP@.5	mAP@.75	mAP@.95
B.	F.	G.			
✓	✓	✓	36.17	21.12	3.89
✓	✓	×	33.83	20.05	3.77
✓	×	×	30.31	17.80	2.82
×	×	×	26.35	15.27	2.66

Table 4. Ablation study for detection performance using top 20 proposals on the ActivityNet validation set. B - Bilinear, F - Flow, G - Global prior

Evaluation Server				
Method	mAP@.5	mAP@.75	mAP@.95	Average
QCIS[36]	42.48	2.88	0.06	14.62
UPC[20]	22.37	14.88	4.45	14.81
UMD[31]	28.67	17.78	2.88	17.68
Oxford[32]	36.40	11.05	0.14	17.83
Ours	37.49	23.47	4.47	23.58

Table 5. Comparison with state-of-the-art methods on the ActivityNet evaluation server using top 20 proposals

and classifier.

Ablation study We conduct a series of ablation studies to evaluate the importance of each component used in our classification model. Table 4 considers three components: "B" stands for "using bilinear pooling"; "F" stands for "using flow" and "G" stands for "using global priors". We can see from the table that each component plays a significant role in improving performance.

Comparison with state-of-the-art We compare our method with state-of-the-art methods [36, 20, 32, 32] submitted during the CVPR 2016 challenge. We submit our results on the evaluation server to measure performance on the test set. At 0.5 overlap, our method is only worse than

[36]. However, this approach was optimized for 0.5 overlap and its performance degrades significantly (to 2%) when mAP at 0.75 or 0.95 overlap is measured. Even though frame level predictions using a Bi-directional LSTM are used in [31], our performance is better when mAP is measured at 0.75 overlap. This is because [31] only performs simple grouping of contiguous segments which are obtained at multiple detection thresholds, instead of a proposal based approach. Hence, it is likely to perform worse on longer action segments.

4.3. The THUMOS14 Dataset

We also evaluate our framework on the THUMOS14 dataset[14], which contains 20 action categories from sports. The validation set contains 1010 untrimmed videos with 200 videos as containing positive samples. The testing set contains 1574 untrimmed videos, where only 213 of them have action instances. We exclude the remaining background videos from our experiments.

Note that solutions for action and activity detection could be different in general, as activities could be very long (minutes) while actions last just a few seconds. Due to their long duration, evaluation at high overlap (0.8 e.g.) makes sense for activities, but not for actions. Nevertheless, we also train our proposed framework on the validation set of THUMOS14 and test on the testing set. Our model also outperforms state-of-the-art methods on proposal metrics by a significant margin, which shows the good generalization ability of our approach.

Performance of our ranker Our proposal ranker outperforms existing algorithms like SCNN[30] and DAPs[6]. We show proposal performance on both average recall calculated using IoU thresholds from 0.5 to 1 at a step 0.05 (shown in Table 6) and recall at IoU 0.5 (shown in Table 7) using 10, 50, 100, 500 proposals. Our proposal ranker performs consistently better than previous methods, especially using small number of proposals.

In Table 8, it is clear that, the proposal ranker performance improves significantly when using a pair of context windows as input. Hence, it is important to use context features for localization in videos, which has been largely ignored in previous state-of-the-art activity detection methods.

Comparison with state-of-the-art Using off the shelf classifiers and our proposals, we also demonstrate noticeable improvement in detection performance on THUMOS14. Here, we compare our temporal context network with DAPs[6], PSDF[15], FG[27] SCNN[30] and CDC[28]. We replace the S-CNN proposals originally used in CDC with our proposals. For scoring the detections in CDC, we multiply our proposal scores with CDC’s classification score. We show that our proposals further benefit CDC and improve detection performance consistently at different

Method	Avg.Recall [0.5:0.05:1]			
	@ 10	@ 50	@ 100	@ 500
DAPs	3.0	11.7	20.1	46.7
SCNN	5.5	16.6	24.8	48.3
Ours	7.7	20.5	29.6	49.2

Table 6. Average Recall from IoU 0.5 to 1 with step size 0.05 for our proposals and other methods on the THUMOS14 testing set

Method	Recall(IoU=0.5)			
	@ 10	@ 50	@ 100	@ 500
DAPs	8.4	29.2	46.9	85.5
SCNN	13.0	35.2	49.6	84.1
Ours	17.1	42.8	59.8	88.7

Table 7. Recall evaluation at IoU 0.5 between our proposals and state-of-the-art methods on THUMOS14 testing set

Method	Avg.Recall@100	mAP@0.5
Ours w/o Context	22.5	20.5
Ours w/ Context	29.6	25.6

Table 8. Evaluation on the influence with and without context on THUMOS14 testing set

Method	mAP@.4	mAP@.5	mAP@.6	mAP@.7
DAPs[6]	—	13.9	—	—
FG[27]	26.4	17.1	—	—
PSDF[15]	26.1	18.8	—	—
SCNN[30]	28.7	19.0	—	—
SCNN+CDC[28]	29.4	23.3	13.1	7.9
Ours+CDC	33.3	25.6	15.9	9.0

Table 9. Performance of state-of-the-art detectors on the THUMOS14 testing set

overlap thresholds.

5. Qualitative Results

We show some qualitative results for TCN, with and without context. Note that only top 5 proposals are shown. The ground truth is shown in blue while predictions are shown in green. It is evident that when context is not used, multiple proposals are present inside or just at the boundary of ground truth intervals. Therefore, although the location is near the actual interval (when context is not used), the boundaries are inaccurate. Hence, when detection metrics are computed, these nearby detections get marked as false positives leading to a drop in average precision. However, when context is used, the proposals boundaries are significantly more accurate compared to the case when context is not used.

6. Conclusion

We demonstrated that temporal context is helpful for performing localization of activities in videos. Analysis was performed to study the impact of temporal proposals in

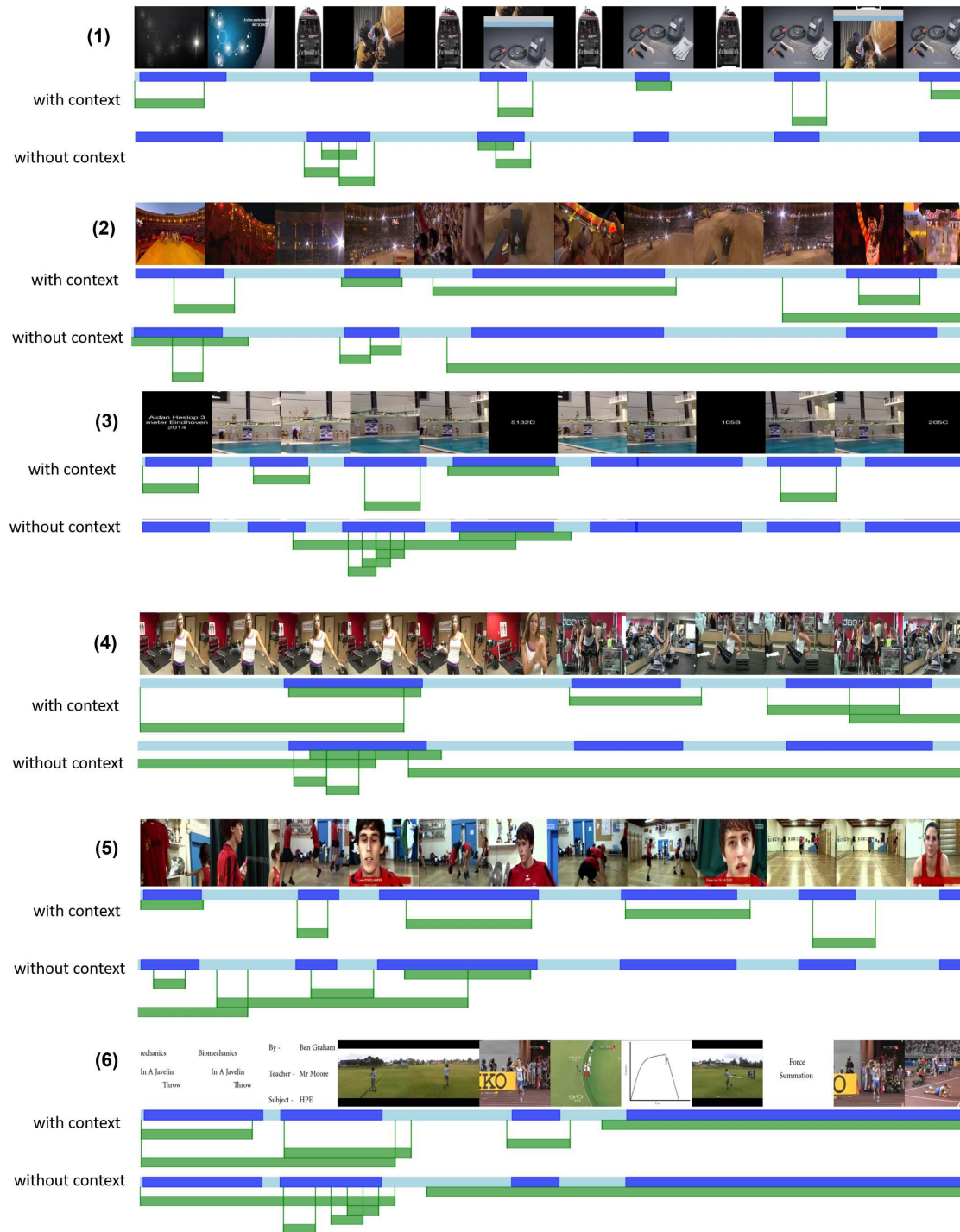


Figure 6. Visualization of top 5 ranking results, the blue bar denotes the ground-truth while the green one represents proposals.

videos by studying precision recall characteristics at multiple overlap thresholds. We also vary the context window to study the importance of temporal context for localization. Finally, we demonstrated state-of-the-art performance on two challenging public datasets.

Acknowledgement

The authors acknowledge the University of Maryland supercomputing resources <http://www.it.umd.edu/hpcc> made available for conducting the research reported in this paper.

References

- [1] I. Atmosukarto, B. Ghanem, and N. Ahuja. Trajectory-based fisher kernel representation for action recognition in videos. In *Pattern Recognition (ICPR), 21st International Conference on*, pages 3333–3336. IEEE, 2012. 2
- [2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):257–267, 2001. 1
- [3] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 568–574. IEEE, 1997. 1
- [4] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1914–1923, 2016. 2
- [5] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 1, 4
- [6] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016. 1, 2, 7
- [7] J. Gemert, M. Jain, E. Gati, C. G. Snoek, et al. *Apt: Action localization proposals from dense trajectories*. BMVA Press, 2015. 2
- [8] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1134–1142, 2015. 3
- [9] G. Gkioxari and J. Malik. Finding action tubes. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 759–768, 2015. 2
- [10] I. Haritaoglu, D. Harwood, and L. S. Davis. W/sup 4: real-time surveillance of people and their activities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):809–830, 2000. 1
- [11] M. Hasan and A. K. Roy-Chowdhury. Context aware active learning of activity recognition models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4543–4551, 2015. 3
- [12] F. C. Heilbron, A. Thabet, J. C. Niebles, and B. Ghanem. Camera motion and surrounding scene appearance as context for action recognition. In *Asian Conference on Computer Vision*, pages 583–597. Springer, 2014. 2
- [13] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 740–747, 2014. 2
- [14] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014. 2, 7
- [15] X. Y. A. A. Jun Yuan, Bingbing Ni. Temporal action localization with pyramid of score distribution features. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2016. 7
- [16] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 817–825, 2016. 3
- [17] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1942–1950, 2016. 2
- [18] P. Mettes, J. C. van Gemert, S. Cappallo, T. Mensink, and C. G. Snoek. Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 427–434. ACM, 2015. 2
- [19] P. S. M. Mettes, D. C. Koelma, and C. G. M. Snoek. The imagenet shuffle: Reorganized pre-training for video event detection. In *ACM International Conference on Multimedia Retrieval*, 2016. 4
- [20] A. Montes, A. Salvador, S. Pascual, and X. Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. In *1st NIPS Workshop on Large Scale Computer Vision Systems*, December 2016. 6
- [21] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 891–898, 2014. 3
- [22] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *European conference on computer vision*, pages 737–752. Springer, 2014. 2
- [23] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150:109–125, 2016. 2
- [24] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV, ECCV’10*, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag. 4
- [25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2
- [26] 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. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2
- [27] G. M. L. F.-F. Serena Yeung, Olga Russakovsky. End-to-end learning of action detection from frame glimpses in videos. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2016. 7
- [28] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise

- temporal action localization in untrimmed videos. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2017. 3, 7
- [29] Z. Shou, D. Wang, and S. Chang. Action temporal localization in untrimmed videos via multi-stage cnns. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2016. 1, 2
- [30] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2016. 7
- [31] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2, 6, 7
- [32] G. Singh and F. Cuzzolin. Untrimmed video classification for activity detection: submission to activitynet challenge. *CoRR*, abs/1607.01979, 2016. 4, 6
- [33] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 3169–3176. 2
- [34] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013. 2
- [35] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016. 4
- [36] R. Wang and D. Tao. Uts at activitynet 2016. *ActivityNet Large Scale Activity Recognition Challenge*, 2016. 6, 7
- [37] Y. Wang and M. Hoai. Improving human action recognition by non-action classification. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2698–2707, 2016. 2
- [38] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 489–496. IEEE, 2011. 3
- [39] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1302–1311, 2015. 2
- [40] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016. 3
- [41] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal action localization with pyramid of score distribution features. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 3093–3102, 2016. 2