DCAN: Improving Temporal Action Detection via Dual Context Aggregation

Guo Chen*, Yin-Dong Zheng*, Limin Wang, Tong Lu†

State Key Lab for Novel Software Technology, Nanjing University, China
{chenguo1177, ydzheng0331}@gmail.com, {lmwang, lutong}@nju.edu.cn

Abstract

Temporal action detection aims to locate the boundaries of action in the video. The current method based on boundary matching enumerates and calculates all possible boundary matchings to generate proposals. However, these methods neglect the long-range context aggregation in boundary prediction. At the same time, due to the similar semantics of adjacent matchings, local semantic aggregation of densely-generated matchings cannot improve semantic richness and discrimination. In this paper, we propose the end-to-end proposal generation method named *Dual Context Aggregation Network* (DCAN) to aggregate context on two levels, namely, boundary level and proposal level, for generating high-quality action proposals, thereby improving the performance of temporal action detection. Specifically, we design the Multi-Path Temporal Context Aggregation (MTCA) to achieve smooth context aggregation on boundary level and precise evaluation of boundaries. For matching evaluation, Coarse-to-Fine Matching (CFM) is designed to aggregate context on the proposal level and refine the matching map from coarse to fine. We conduct extensive experiments on ActivityNet v1.3 and THUMOS-14. DCAN obtains an average mAP of 35.39% on ActivityNet v1.3 and reaches mAP 54.1% at IoU@0.5 on THUMOS-14, which demonstrates DCAN can generate high-quality proposals and achieve state-of-the-art performance. We release the code at https://github.com/cg1177/DCAN.

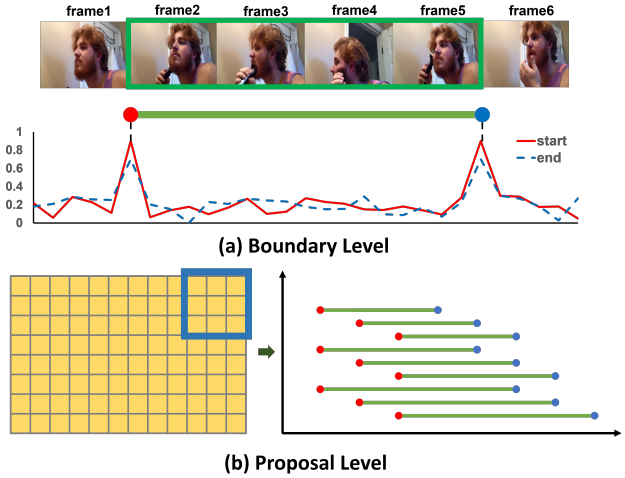

Figure 1: Examples to illustrate the importance and difficulty of context aggregation on two levels. (a) The frame_2 and frame_5 with similar local context are difficult to be distinguished, which will generate many positive false proposals. (b) The temporal intervals of adjacent 3×3 matchings on the matching map are highly overlapped, lacking semantic discrimination and richness, and it is not easy to obtain effective semantic supplements by simple aggregation.

Indroduction

With the explosive growth of Internet videos, video content's understanding and analysis technology have attracted more academia and industry attention. Temporal action detection is to locate the boundaries of action instances and recognize action categories in untrimmed videos. Video data is a stack of image frames, and the semantic changes between frames are more complicated to capture than the semantic changes between image pixels. Therefore, compared with object detection, temporal action detection focuses more on processing and capturing the temporal information of the video.

Similar to object detection, the temporal action detection methods can be divided into one-stage and two-stage methods. The one-stage methods simultaneously locate the action boundaries and classify the action. The two-stage methods first generate proposals, then refine the boundaries and classify the action. In order to obtain high-quality detection results, the proposals should precisely cover action with high recalls and reliable confidence scores. There are mainly two types of proposal generation methods, "top-down" and "bottom-up". The "top-down" method (Gao et al. 2017; Long et al. 2019; Gao et al. 2020; Chao et al. 2018; Lin, Zhao, and Shou 2017) obtains the final proposals by refining the boundaries of anchors or sliding windows with predefined scales and calculating the confidence. The "bottom-up" method (Zhao et al. 2017a; Lin et al. 2018, 2019; Bai et al. 2020; Su et al. 2021; Xu et al. 2020) generates proposals by calculating the boundary confidence of each position and matching start positions with end positions.

*These authors contributed equally.
†Corresponding author.

The above "bottom-up" methods generate proposals by the confidence scores of boundaries and matching maps. However, there are some difficulties on boundary-level and proposal-level context aggregation with this framework. First, on the boundary level, different actions vary at different speeds. The boundary of a slow action usually is not a clear temporal position but a transitional interval. So there is not enough semantic information for the precise predictions of these boundaries without effective local temporal context aggregation. On the other hand, as shown in Figure 1(a), the start and end boundaries of some actions are so similar that the high confidence of start and end at such positions will generate many invalid matchings without long-range temporal context aggregation. Second, on the proposal level, it is not proper to simply perform context aggregation on the matching map. As shown in Figure 1(b), aggregating adjacent matchings with different temporal scales and semantic densities will damage the internal semantic representation of the matchings. Moreover, adjacent matchings are highly overlapped so that their semantic information is too similar to obtain sufficient semantic supplement after aggregation. Therefore, it is necessary to design effective context aggregation methods on the temporal and proposal levels.

To mitigate the above issues, we propose a novel method called *Dual Context Aggregation Network* (DCAN) for improving temporal proposal generation. Similar to BMN (Lin et al. 2019), DCAN has a temporal evaluation branch and a matching evaluation branch. For the temporal evaluation branch, we design the *Multi-Path Temporal Context Aggregation* (MTCA) to achieve effective and smooth context aggregation on the boundary level. MTCA is a stack of Multi-Path Temporal Convolutions (MPTC). In each MPTC, there is a long-range path equipped with a dilated convolution to expand the receptive field and achieve long-range context aggregation and a short-range path with a regular convolution to aggregate short-range context. In order to alleviate the gridding artifacts of dilated convolution, we adopt a saw-tooth wave-like heuristic arrangement for MPTCs to ensure the context of each position can be aggregated smoothly. MTCA gradually expands the receptive field from the frame to the entire video, thereby effectively aggregating the long-range and short-range contexts. For the matching evaluation branch, we propose the *Coarse-to-fine Matching* (CFM) for effective context aggregation on the proposal level. CFM first generates a coarse matching map using the Group Sampling strategy, then gradually refines the map from coarse to fine through the refinement network. The coarse map ensures the distinction of semantic information between sparse matchings, and at the same time, aggregates the context of adjacent matchings without damaging the semantic representation. During the coarse-to-fine process, the relation between matchings is gradually supplemented and restored. CFM enhances the expressiveness and robustness of the matching context, and the final matching map contains the relation between the matchings.

We conduct extensive experiments on the THUMOS-14 and ActivityNet v1.3 to demonstrate the effectiveness of our Dual Context Aggregation Network (DCAN). In summary, our contributions are as follows:

- On the boundary level, we propose the Multi-Path Temporal Context Aggregation to aggregate boundaries context and alleviate the gridding artifacts of dilated convolutions.
- On the proposal level, we design the Coarse-to-fine Matching to generate and refine matching maps from coarse to fine, which enhances the expressiveness and robustness of the matching context.
- The experiments prove the high performance of DCAN, which can achieve better performance than other state-of-the-art methods on ActivityNet v1.3 and THUMOS-14.

Related Work

Action Recognition

Action recognition is an essential task in video understanding. It performs spatio-temporal modeling on video frames to recognition actions in the video. 2D CNN methods (Simonyan and Zisserman 2014; Wang et al. 2016; Lin, Gan, and Han 2019; Feichtenhofer et al. 2019; Liu et al. 2020; Li et al. 2020; Wang et al. 2021) take RGB and optical flow as input and perform spatial and temporal modeling, respectively. 3D CNN methods (Tran et al. 2015, 2017; Carreira and Zisserman 2017; Diba et al. 2017) capture spatio-temporal information between frames by performing 3D convolution on stacked video frames. (Qiu, Yao, and Mei 2017; Xie et al. 2018; Tran et al. 2018) model spatio-temporal features by decoupling 2D and 1D convolutions for reducing computing resources. In this work, we use 2-stream (Simonyan and Zisserman 2014) to generate the feature sequence of the video.

Temporal Action Detection and Proposals

Current temporal action detection methods can be divided into one-stage and two-stage methods. The one-stage methods simultaneously generate action proposals and corresponding action labels in a single model. The two-stage methods first generate the proposals, then refine the boundaries and classify actions. (Gao et al. 2017, 2020; Chao et al. 2018) generate proposals based on a pre-defined sliding window or anchors and train a classifier to filter anchors. TURN (Gao et al. 2017) utilizes a sliding window and refines boundaries by concatenating the boundary context and internal context of proposals. RapNet (Gao et al. 2020) proposes a relation-aware module to capture long-range context between frames. RTD-Net (Tan et al. 2021) utilizes the transformer decoder to generate a sparse proposal set directly, effectively omitting post-processing steps. Although many anchor-based methods use multi-scale anchors to increase the diversity, the generated proposals are still not flexible enough to cover actions of varying temporal scales. (Lin et al. 2018, 2019; Su et al. 2021) use a flexible way called boundary matching. They predict each frame's start and end confidence, then match the frames with high start and end confidences to generate the proposals and evaluate their confidence. These methods are more difficult to optimize due to the lack of prior knowledge of the anchor.

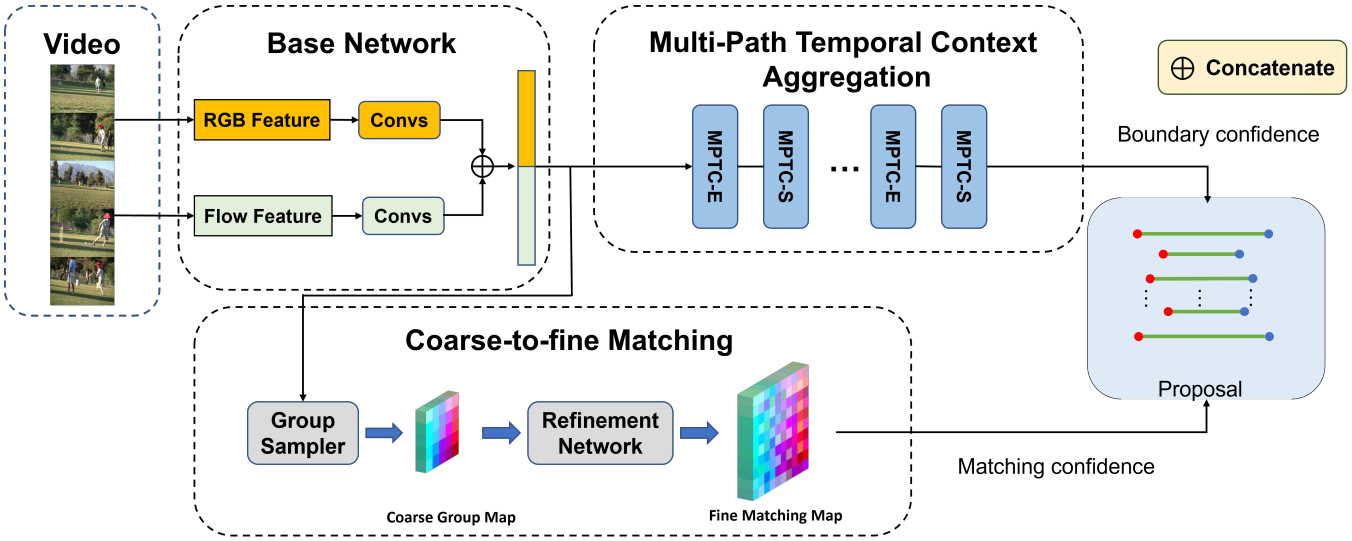

Figure 2: The network architecture of DCAN. First, a dual-path convolution layer is used to model local temporal features with RGB and flow features, respectively. Then, we concatenate these two features and feed them into Multi-Path Temporal Context Aggregation Module to aggregate temporal context for boundary confidence evaluation. At the same time, these features are also input into Coarse-to-fine Matching Module, which generates a coarse group map and then refines the map to a fine matching map through a refinement network for matching confidence evaluation. Finally, the boundary and matching confidence will be combined to obtain final proposals.

Temporal Modeling in Temporal Action Detection

Temporal modeling plays an important role in temporal action detection. (Escorcia et al. 2016; Yeung et al. 2016) use LSTM to generate action proposals. Compared with LSTM, 1D convolution on temporal dimension shows better performance when modeling the long-range temporal structure of actions. (Lea et al. 2017; Gong, Zheng, and Mu 2020) and (Su et al. 2021) utilizes temporal convolution and UNet for temporal relationship modeling, respectively. (Qing et al. 2021) aggregates local and global temporal context by two types of self-attention modules. We use stacked Multi-Path Temporal Convolution to capture the long-term and short-term dependence of the frames.

Approach

Problem Definition

For an untrimmed video, we denote it as $U = \{u_t\}_{t=1}^{l_v}$, where l_v indicates the length of the video and u_t is the t-th frame. We denote the temporal annotation of action instances as $\Psi_g = \{\varphi_n = (t_s, t_e)\}_{n=1}^{N_g}$ in the video S_v which has N_g instance. t_s and t_e are the start and end boundaries of the instance φ respectively. The action detection model generates prediction proposals that should cover Ψ_g with high recall and high temporal overlapping.

Overview of DCAN

As shown in Figure 2, DCAN is composed of three modules: Base Network, Multi-Path Temporal Context Aggregation Module, and Coarse-to-fine Matching Module. Firstly, the video frames are fed into the Base Network for local temporal modeling. Then the features enter into Multi-Path Temporal Context Aggregation Module and Coarse-to-fine Matching Module to perform the boundary-level and proposal-level context aggregation, respectively. The aggregated feature will be used for boundary and matching evaluation, finally generating the proposals. We present the technical details of three modules in the following sections.

Base Network

Following recent proposal generation methods (Lin et al. 2018, 2019), we take the temporal features which are extracted using the action recognition backbone network with a fixed-interval sliding window as input. This feature extracting method only extracts the semantic feature of local temporal sequence frames in an isolated window, resulting in a lack of correlation between adjacent windows, so we design a base network to perform local context aggregation on the temporal feature. The base network has two convolution paths to model RGB features and optical flow features, respectively. Each path consists of N_{base} 1D convolution layers, with 128 filters, kernel size of 3, stride of 1, followed by a ReLU activation layer. Finally, we obtain the feature $F_{\text{base}}^{\text{rgb}}$ and $F_{\text{base}}^{\text{flow}}$, then feed them to the following modules.

Multi-Path Temporal Context Aggregation

In this section, we introduce our Multi-Path Temporal Context Aggregation (MTCA) to aggregate long-range and short-range temporal context for temporal evaluation effectively. As shown in Figure 2 and Figure 3, MTCA is com-

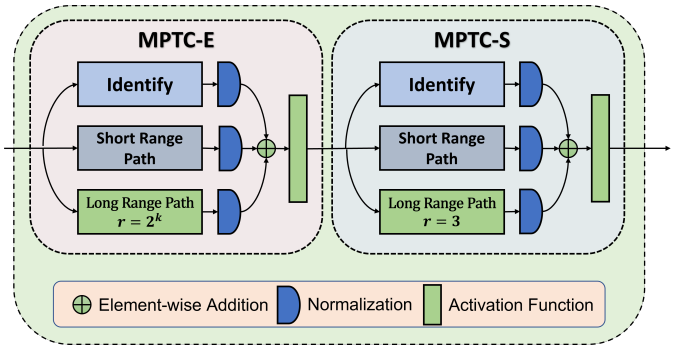

Figure 3: The architecture of MPTC. First, temporal features are fed into MPTC-E with an increasing dilation to expand the receptive field. Then, MPTC-S with a fixed dilation smooths the features from MPTC-E.

posed of a sequence of stacked Multi-Path Temporal Convolution (MPTC). In each MPTC, $\mathrm{Conv_L}$ is the long-range path, including a dilated convolution layer with a kernel size of 3 and a dilation of r to aggregate long-range context and extend the receptive field. $\mathrm{Conv_S}$ is the short-range path including a regular convolution with a kernel size of 3 that aggregates short-range temporal context. In order to enhance the representation ability of features, and at the same time, solve network degradation during training, we introduce a shortcut path to fuse features of different levels. Finally, we combine these three paths in parallel and do the element-wise addition. MPTC can be formulated as follows:

$$\mathrm{MPTC}(x) = \sigma(\varepsilon(\mathrm{Conv_L}(x)) + \varepsilon(\mathrm{Conv_S}(x)) + \varepsilon(x)), \quad (1)$$

where $\sigma(\cdot)$ and $\varepsilon(\cdot)$ is the nonlinear activation function and the normalization operation respectively.

Previous research (Wang et al. 2018) on the application of dilated convolution in image detection and segmentation tasks proves that too-rapid expansion of the receptive field may lead to information loss. Specifically, simply stacking multiple dilated convolution layers with exponentially increased dilation will cause gridding artifacts, that is, features at some positions cannot participate in the calculation. To alleviate this phenomenon, we design two different types of MPTC. The first, called MPTC-E, is an MPTC with dilation of 2^k for rapidly expanding the receptive field, where k is the exponent to adjust the scale of receptive field expansion and increases with the depth of the network increases. The second, called MPTC-S, is an MPTC with a fixed dilation of r_{smooth} for alleviating the gridding artifacts.

The numbers of MPTC-E and MPTC-S in MTCA are both N_{b}, and MPTC-E's dilation r_i is 2^i where $1 \leq i \leq N_{\mathrm{b}}$. To alleviate the gridding artifacts as much as possible, we set the dilations of MPTC-E and MPTC-S are relatively prime and connect these two blocks alternately. In this way, the top layer of MTCA can access information from the entire video while the aggregation of information at each temporal position is smooth and uniform. Finally, after aggregating long-range and short-range context at each temporal position, the start and end probabilities are predicted, respectively.

Discussion. Some methods use the self-attention mechanism to enhance and aggregate the features of each position, but we argue that the self-attention mechanism is not suitable for boundary evaluation. The self-attention mechanism pays more attention to the correlation between positions but ignores the order and distance. For boundary evaluation, context aggregation around the boundary is more valuable than aggregation at a distance. We hope that the position can focus on aggregating the information of the action instance to which it belongs while also taking into account the aggregation of long-range context rather than the uniform aggregation of the positions of the entire video. Therefore, MTCA is more beneficial to long-range and short-range context aggregation than the self-attention mechanism.

Coarse-to-fine Matching

Coarse-to-fine Matching (CFM) aims to achieve proposal-level context aggregation by constructing a coarse matching map and refining the map from coarse to fine. Since adjacent matchings in the matching map have similar sampling intervals and sampling points, which leads lack of distinction and richness in semantics. Context aggregation of such adjacent matchings cannot obtain an effective semantic supplement. At the same time, aggregating adjacent matchings with different temporal scales and different semantic densities on the matching map will damage the semantic representation inside each matching. Hence we propose the Group Sampling strategy to construct the group map. Specifically, the matchings in the range of $G \times G$ on the original matching map (Lin et al. 2019) are grouped into a group, and we take the union of their temporal intervals to construct the features of the group. The entire original matching map can be divided into $\frac{D}{G} \times \frac{T}{G}$ groups without overlap, where T and D is the scale of temporal feature and the maximum duration of any possible action instance. The strategy can be formulated by:

$$P_{i,j} = \mathrm{GroupSample}(F_{\mathrm{p}}, s_{i,j}, e_{i,j}), \quad (2)$$

where $\mathrm{GroupSample}(F, s, e)$ operation is to uniformly sample N_{sample} points from position s ($0 \leq s < e \leq 1.0$) to position e of temporal feature F. $P_{i,j} \in R^{128 \times N_{\mathrm{sample}}}$ is the sampled group feature of the i-th row and j-th column on the group map. We use hyper-parameter G to set group size and obtain the indices of the group map:

$$0 \leq i < \frac{D}{G},\ 0 \leq j < \frac{T}{G}. \quad (3)$$

The corresponding start position $s_{i,j}$ and end position $e_{i,j}$ of sampling and is also formulated by:

$$s_{i,j} = \frac{j \times G}{T},\ e_{i,j} = s_{i,j} + \frac{(i+1) \times G}{T}. \quad (4)$$

Through the above method, we can obtain each group's start and end boundaries and then use the feature construction method of BMN to generate sample-level group map $M_{\mathrm{g}} \in R^{\frac{D}{G} \times \frac{T}{G} \times 128 \times N_{\mathrm{sample}}}$, where 128 is the dimension of $P_{i,j}$. We obtain the group map $M'_{\mathrm{g}} \in R^{\frac{D}{G} \times \frac{T}{G} \times C}$ using a linear transformation to M_g.

Then, we refine the coarse group map $M_{\mathrm{g}}^{'}$ to the fine matching map $M_{\mathrm{m}} \in R^{D \times T \times C}$ using a refinement network. The refining process has two steps. Firstly, we adopt deconvolutions to upsample $M_{\mathrm{g}}^{'}$ that each deconvolution layer upsamples the map with factor 2 on the temporal and duration dimensions, and each group feature is finally refined to $G \times G$ matching features. Then, a convolution with the kernel size of 3 is adopted to rebuild the relationship of adjacent matchings. While the refinement network gradually refines group features to matching features, it also restores the relation between matchings and realizes the implicit aggregation of context between matchings.

Discussion. The final convolution operation is the same as BMN (Lin et al. 2019), but their effects are different. Our matching features are refined from the group features. The construction of the group features weakens the internal temporal scale representation, so it can be considered that the internal temporal semantic representations of different temporal scales matchings are homogeneous. Performing convolution on such features has a smoother context aggregation effect and can better capture the internal relationship between matchings, so its effect is better than PEM of BMN.

Training and Inference For DCAN

Training

We train DCAN in the form of a multitask loss function, including a boundary classification loss L_{b}, a proposal evaluation loss L_{p} and a regularization where β is set to 0.0001:

$$L = L_{\mathrm{b}} + L_{\mathrm{p}} + \beta \cdot L_2(\Theta). \quad (5)$$

L_{b} is used to classify whether each frame is the start position or end position of the action:

$$L_{\mathrm{b}} = L_{\mathrm{WCE}}(P^{\mathrm{start}}, G^{\mathrm{start}}) + L_{\mathrm{WCE}}(P^{\mathrm{end}}, G^{\mathrm{end}}), \quad (6)$$

where L_{WCE} is the weighted binary cross-entropy loss function, P^{start} is the predicted start probability of frames (same for P^{end}), G^{start} and G^{end} are the binary ground-truth which are obtained by calculating IoR between action instances and temporal positions. The loss L_{p} is used to evaluate proposals confidence. Following BMN, we predict two confidence map M^{cls} and M^{reg}, which are trained by the weighted binary cross-entropy loss function and mean squared error loss function respectively:

$$L_p = L_{\mathrm{WCE}}(M^{\mathrm{cls}}, G^{\mathrm{cls}}) + \lambda \cdot L_2(M^{\mathrm{reg}}, G^{\mathrm{IoU}}), \quad (7)$$

where G^{IoU} is the IoU map calculated by proposals and ground truth, G^{cls} is the foreground-background map obtained by binarizing G^{IoU} with a threshold 0.9, and λ is the loss weight, which is set to 10 as default in our experiments.

Inference

Score Fusion. During the inference stage, the temporal evaluation branch generates the start probability P^{start} and the end probability P^{end} for each position. We use these two probabilities as the two boundaries scores of the proposals and fuse them with the matching score maps M^{cls} and M^{reg} obtained by the matching evaluation branch to generate the final score of proposals. Take the proposal $\varphi = [t_{\mathrm{s}}, t_{\mathrm{e}}]$ for example, the combination of final score p_φ can be shown as:

$$p_\varphi = P^{\mathrm{start}}_{t_{\mathrm{s}}} \cdot P^{\mathrm{end}}_{t_{\mathrm{e}}} \cdot (M^{\mathrm{cls}}_{t_{\mathrm{e}}-t_{\mathrm{s}},t_{\mathrm{s}}} \cdot M^{\mathrm{reg}}_{t_{\mathrm{e}}-t_{\mathrm{s}},t_{\mathrm{s}}})^{\gamma}, \quad (8)$$

where γ is a hyperparameter for adjusting the compatibility of boundary scores and matching scores and is set as 1.5 on THUMOS-14 and 0.8 on ActivityNet v1.3, respectively.

Post Processing. After score fusion, DCAN generates the proposal candidates set as $\Psi_{\mathrm{c}} = \{\varphi_n = (t_{\mathrm{s}}, t_{\mathrm{e}}, p)\}_{n=1}^{N_{\mathrm{c}}}$ and then we post-process the candidate proposals to remove redundant proposals. First, we remove proposals whose start or end probability is lower than half of the corresponding maximum value. Then, we adopt the Soft-NMS (Bodla et al. 2017) to eliminate the redundant proposals and obtain the final proposals set $\Psi_{\mathrm{final}} = \{\varphi_n = (t_{\mathrm{s}}, t_{\mathrm{e}}, p)\}_{n=1}^{N_{\mathrm{final}}}$, where the number of final proposals is N_{final}.

Experiments

Datasets and Setup

THUMOS-14. (Jiang et al. 2014) contains 413 temporal annotated untrimmed videos with 20 action categories. We use 200 videos in the validation set for training and 213 videos in the testing set for evaluation.

ActivityNet v1.3. (Heilbron et al. 2015) is a large-scale action understanding dataset, which consists of 19,994 videos for training, 4,728 for validation, and 5,044 for testing, with 200 action classes. The total duration of the videos is about 600 hours. ActivityNet v1.3 only contains 1.5 occurrences per video on average, and most videos contain a single action category with 36% background on average.

Evaluation Metrics. Average Recall (AR) is the average recall under specified tIoU thresholds for measuring the quality of proposals. On ActivityNet v1.3, these thresholds are set to $[0.5 : 0.05 : 0.95]$. On THUMOS-14, they are set to $[0.5 : 0.05 : 1.0]$. Limiting the average number (AN) of proposals for each video allows us to calculate the area under the AR vs AN curve to obtain AUC. On ActivityNet v1.3, we set AN from 1 to 100. The quality of temporal action detection requires evaluating mean Average Precision(mAP) under multiple tIoU. On ActivityNet v1.3, the tIoU thresholds are set to $\{0.5, 0.75, 0.95\}$, and we also test the average mAP of tIoU thresholds between 0.5 and 0.95 with the step of 0.05. On THUMOS-14, these tIoU thresholds are set to $\{0.3, 0.4, 0.5, 0.6, 0.7\}$.

Implementation Details. We use pre-extracted features for all datasets and train the network from scratch. For ActivityNet v1.3, we adopt the two-stream network (Xiong et al. 2016) fine-tuned on the training set of ActivityNet v1.3 with frame interval $\sigma = 16$. Each video feature sequence is rescaled to $L = 100$ snippets using linear interpolation. We set the batch size to 16 and the learning rate to 0.001 for the first 7 epochs and 0.0001 for the following 3 epochs. For THUMOS-14, the features are extracted using TSN (Wang et al. 2016) pre-trained on Kinetics (Kay et al. 2017) with $\sigma = 5$. We crop each video feature sequence with overlapped windows of size $L = 256$ and stride 128. In training,

Method	@50	@100	@200	@500	@1000
TAG	18.55	29.00	39.61	-	-
CTAP	32.49	42.61	51.97	-	-
BSN	37.46	46.06	53.23	61.35	65.10
MGG	39.93	47.75	54.65	61.36	64.06
BMN	39.36	47.72	54.84	62.19	65.49
BSN++	42.44	49.84	57.61	**65.17**	66.83
TCANet	42.05	50.48	57.13	63.61	66.88
DCAN	**42.65**	**51.05**	**57.95**	64.58	**68.37**

Table 1: Comparison of DCAN with other state-of-the-art methods on THUMOS-14 in terms of AR@AN. All models use the two-stream feature as input.

Method	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**
TURN	6.3	14.1	24.5	35.3	46.3
BSN	20.0	28.4	36.9	45.0	53.5
MGG	21.3	29.5	37.4	46.8	53.9
BMN	20.5	29.7	38.8	47.4	56.0
G-TAD	23.4	30.8	40.2	47.6	54.5
BSN++	22.8	31.9	41.3	49.5	59.9
RTD-Net	25.0	36.4	45.1	53.1	58.5
TCANet	26.7	36.8	44.6	53.2	60.6
DCAN	**32.6**	**43.9**	**54.1**	**62.7**	**68.2**

Table 2: Comparison between DCAN with other state-of-the-art methods on THUMOS-14 dataset. The results are measured by mAP(%) at different tIoU thresholds. Proposals from all methods are combined with video-level classifier UntrimmedNet (Wang et al. 2017).

we do not use any clips void of actions. We set the batch size to 16 and the learning rate to 0.0004 for all 5 epochs.

The N_{b} is set to 6 on ActivityNet v1.3 and 7 on THUMOS-14. The N_{base}, N_{sample}, r_{smooth} and G are set to 3, 32, 3 and 2. In the post-processing, the Soft-NMS threshold is set to 0.5 to pick the top N_{final} confident predictions, where N_{final} is 100 for ActivityNet v1.3 and 200 for THUMOS-14.

Comparison with State-of-the-art Results

THUMOS-14. We compare DCAN with other state-of-the-art methods on THUMOS-14 in Table 1 and Table 2, where DCAN significantly improves the performance of the proposal generation and action detection. We report AR@AN for proposal generation and mAP for action detection. As shown in Table 1, DCAN improves the AR of all proposals for proposal generation except for @500. Furthermore, for action detection in Table 2, DCAN can also obtain at least 5.0% improvement when tIoU at any threshold compared to all previous methods.

ActivityNet v1.3. We compare DCAN with the other methods with the state-of-the-art methods on ActivityNet v1.3 in Table 3 and Table 4. We report the AR@AN and AUC for proposal generation and mAP for action detection.

Method	CTAP	BSN	MGG	BMN	DCAN
AR@1	-	32.17	-	-	**34.42**
AR@100	73.17	74.16	74.54	75.01	**75.71**
AUC	65.72	66.17	66.43	67.10	**67.93**

Table 3: Comparison with other state-of-the-art methods CTAP (Gao, Chen, and Nevatia 2018), BSN (Lin et al. 2018), MGG (Liu et al. 2019), BMN (Lin et al. 2019) on ActivityNet v1.3 in terms of AR@AN and AUC.

Method	**0.5**	**0.75**	**0.95**	**Average**
SSN	39.12	23.48	5.49	23.98
BSN	46.45	29.96	8.02	30.03
RTD-Net	47.21	30.68	8.61	30.83
BMN	50.07	34.78	8.29	33.85
G-TAD	50.36	34.60	9.02	34.09
BC-GNN	50.56	34.75	9.37	34.26
BSN++	51.27	35.70	8.33	34.88
DCAN	**51.78**	**35.98**	**9.45**	**35.39**

Table 4: Comparison between DCAN with other state-of-the-art methods on ActivityNet v1.3. The results are measured by mAP(%) at different tIoU thresholds and average mAP(%). We combined our proposals with video-level classification results from (Zhao et al. 2017b).

For a fair comparison, in the proposal generation task, we only compare the methods without the re-sampling strategy. On two tasks, DCAN outperforms the other state-of-the-art proposal generation methods. Since DCAN improves AR@1 to 34.42, which demonstrates that the proposals with high confidence also have high recalls, bringing a significant performance improvement in the action detection task.

Ablation Studies

Ablation comparison with other "bottom-up" methods. We conduct a direct comparison to other "bottom-up" methods, namely, BSN, BMN, and BSN++ in Table 5 to confirm the effectiveness and superiority of DCAN.

In the temporal evaluation phase, the TEM of BSN and BMN, which only considers the local details for boundary evaluation, is inferior with limited receptive fields for boundary-level context aggregation. BSN++ adopts a shallow U-shaped network to expand the temporal receptive field, but it does not expand the receptive field to global. MTCA realizes the long-range and short-range smooth context aggregation and expands the receptive field to the entire video. Therefore, only replacing the TEM with MTCA improves the mAP of ActivityNet v1.3 and THUMOS-14 from 33.85% and 38.80% to 35.02% and 52.55%.

The BMN and BSN++ directly generate a dense matching map. This matching feature construction method causes the semantics of adjacent matchings on the map to be similar as well as lack distinction and richness. In addition, these methods aggregate proposal-level context by applying convolutions or self-attention modules, ignoring that aggregat-

Model	TEB	MEB	ANet-1.3	THU-14
BSN	TEM	PEM	30.03	36.90
BMN	TEM	PEM	33.85	38.80
BSN++	CBG	PRB	34.88	41.30
DCAN	MTCA	PEM	35.02	52.55
DCAN	TEM	CFM	34.80	49.42
DCAN	MTCA	CFM	**35.39**	**54.14**

Table 5: Ablation study results on the validation set of ActivityNet v1.3 and the test set of THUMOS-14. TEB and MEB denote the modules in Temporal Evaluation Branch and Matching Evaluation Branch. We show experimental results on ActivityNet v1.3 in terms of average mAP(%) and on THUMOS-14 in terms of mAP@0.5(%).

Dataset	mAP	$*$	$r=1$	$r=3$	$r=5$
ActivityNet	0.5	51.49	51.56	51.78	**51.86**
	0.75	35.61	35.92	35.98	**36.00**
	0.95	8.55	8.87	**9.45**	8.86
	Avg	35.01	35.18	**35.39**	35.28
THUMOS-14	0.3	65.16	66.80	**68.21**	67.65
	0.4	59.40	61.59	**62.73**	61.36
	0.5	51.03	53.67	**54.14**	53.19
	0.6	41.40	42.17	**43.91**	42.90
	0.7	30.58	32.11	**32.62**	31.48

Table 6: The effect of different r_{smooth} of the MPTC-S on ActivityNet v1.3 and THUMOS-14 in terms of mAP(%).

ing proposals with multiple different temporal scales and different semantic densities will damage the semantic representation inside the matchings. CFM refines the coarse group map into the fine matching map. The semantic distinction and richness between adjacent matchings are better, and the temporal scale representation inside the matching is weakened, so the context aggregation is more effective. After equipping BMN with CFM, the mAPs of the two datasets are increased by 0.95% and 10.62%, respectively.

When MTCA and CFM work together, the results of DCAN reach 35.39% and 54.41%, which fully demonstrates the importance of context aggregation at two levels and the effectiveness of MTCA and CFM.

How to choose r_{smooth} of the MPTC-S? We conduct experiments to explore how to choose r_{smooth} of MPTC-S for smoothing the receptive field. In order to alleviate the gridding artifacts caused by dilated convolution, the dilation of MPTC-S must be mutually prime with the dilation of MPTC-E. Since the dilation of MPTC-E is set as 2^k, we choose 1, 3 and 5 as the candidate dilation of MPTC-S. The experimental results are shown in Table 6. We also give the result without MPTC-S and denote it as $*$. The experimental results show that MPTC-S can improve performance, and the performance is best when $r_{\text{smooth}} = 3$.

G	0.5	0.75	0.95	Average
1	51.34	35.82	8.90	35.11
2	**51.78**	**35.98**	**9.45**	**35.39**
4	51.51	35.71	8.83	35.01

Table 7: The effect of different group size G of CFM on ActivityNet v1.3 dataset in terms of mAP(%).

What is the effect of group size G of CFM? We explore the different group size G of CFM, and the experimental results are shown in Table 7. When $G = 1$, CFM degenerates into the dense matching map similar to BMN. For $G = 2$ and $G = 4$, we use one-layer and two-layer refinement networks to refine the map from coarse to fine, respectively. The experimental results demonstrate that $G = 2$ is the best choice. $G = 4$ does not work because when the group range is too large, and the generated map is too coarse, encoding 4×4 proposals into 1 grouped matchings will lead to the loss of semantic information, and it is difficult to restore this part of the lost information through refinement.

Efficiency Analysis As an End-to-End Model, it is critical to have excellent inference speed and throughput. Table 8 shows the performance advantages of our network compared with other networks. The inference speed of DCAN is close to BMN, but it has gained a 1.52% improvement on mAP.

Model	TEB	MEB	mAP	T_{total}
BSN	TEM	PEM	30.03	0.629
BMN	TEM	PEM	33.85	0.052
DCAN	MTCA	PEM	35.02	0.053
DCAN	TEM	CFM	34.80	0.047
DCAN	MTCA	CFM	35.39	0.051

Table 8: The efficiency analysis on ActivityNet v1.3. Inference speed here is the seconds (s) cost T_{total} for processing a 3-minute video using an Nvidia 1080-Ti graphics card.

train	Seen(val)		Unseen(val)	
	AR@100	AUC	AR@100	AUC
Seen+Unseen	74.34	66.65	75.55	67.76
Seen	73.43	65.10	**72.58**	**64.86**

Table 9: Generalizability evaluation on ActivityNet v1.3.

Generalizability. To evaluate the generalizability of proposals, we follow BMN and choose two different subsets on ActivityNet v1.3 for generalization ability analysis. The experiment extracts two un-overlapped action subsets from ActivityNet v1.3: "Sports, Exercise, and Recreation" as seen subsets and "Sports, Exercise, and Recreation" as unseen subsets separately. Table 9 shows the results of DCAN on 2stream features. The results reveal that there is only a slight performance drop on unseen categories, suggesting

that DCAN achieves great generalizability to generate high-quality proposals for unseen actions.

Visualization

Comparison of boundary probabilities. We visualize the start and end probability sequences of BMN's TEM and MTCA in Figure 4. It can be observed that for some boundary positions, TEM is difficult to distinguish whether they are the start or the end boundaries, and some non-boundary positions are recognized as boundaries, which demonstrates only using the local context is difficult to evaluate temporal boundary. The probability curve of MTCA is smoother and more distinguishable than TEM, and the probabilities of non-boundary position are significantly lower than those of boundaries. This indicates that long-range and short-range context aggregation on the boundary level can improve the model's ability to distinguish confusing positions and suppress non-boundary positions.

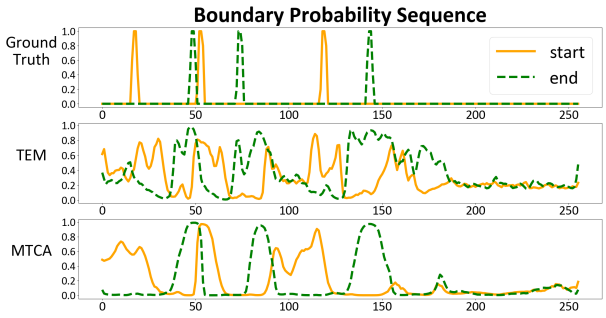

Figure 4: The start and end probability sequences generated by TEM and MTCA.

Comparison of matching maps. We visualize and compare the matching maps generated by CFM. First, to verify the effectiveness of CFM, we visualize two matching maps obtained by $G = 1$ and $G = 2$ in Figure 5. When $G = 1$, the matching map degenerates into the dense matching map similar to BMN. Compared with $G = 1$, $G = 2$ has more precise map boundaries and less noise, proving that CFM solves the lack of distinction and richness to a certain extent. Then, to prove that the process from coarse to fine is necessary for the matching map, we visualize the coarse and fine matching map. As shown in Figure 6, the G in experiments is 2, and the resolutions of coarse and fine matching maps are 50×50 and 100×100, respectively. Compared with the features of the coarse map, the high-response area of the fine matching map has significantly shrunk, indicating that the process from coarse to fine can effectively suppress the high-response area with lower tIoU, thereby improving the credibility of the final proposal score.

Conclusion

In this paper, we have proposed a novel Dual Context Aggregation Network (DCAN) for high-quality proposal generation and action detection. DCAN aggregating context on boundary and proposal level, respectively. On the boundary level, Multi-Path Temporal Context Aggregation (MTCA) uses multiple paths to aggregate long-range and short-range context smoothly. Coarse-to-fine Matching (CFM) refines the matching map from coarse to fine and achieves effective context aggregation on the proposal level. Extensive experiments on ActivityNet v1.3 and THUMOS-14 demonstrate two-level context aggregation can significantly improve proposal generation and action detection performance.

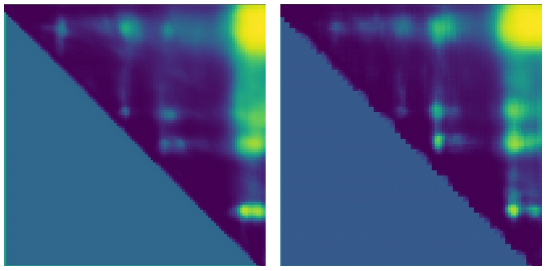

Figure 5: Two dense matching maps obtained by $G = 1$(left) and $G = 2$(right).

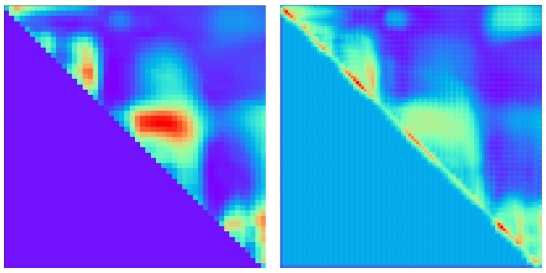

Figure 6: The features of the coarse(left) and the fine(right) matching map

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61672273, No. 61832008, No. 62076119, No. 61921006), Program for Innovative Talents and Entrepreneur in Jiangsu Province, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Bai, Y.; Wang, Y.; Tong, Y.; Yang, Y.; Liu, Q.; and Liu, J. 2020. Boundary Content Graph Neural Network for Temporal Action Proposal Generation. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J., eds., *ECCV*, volume 12373, 121–137.

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Soft-NMS - Improving Object Detection with One Line of Code. In *ICCV*, 5562–5570.

Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 4724–4733.

Chao, Y.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R. 2018. Rethinking the Faster

R-CNN Architecture for Temporal Action Localization. In *CVPR*, 1130–1139.

Diba, A.; Fayyaz, M.; Sharma, V.; Karami, A. H.; Arzani, M. M.; Yousefzadeh, R.; and Gool, L. V. 2017. Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification. *CoRR*, abs/1711.08200.

Escorcia, V.; Heilbron, F. C.; Niebles, J. C.; and Ghanem, B. 2016. DAPs: Deep Action Proposals for Action Understanding. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *ECCV*, volume 9907, 768–784.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. SlowFast Networks for Video Recognition. In *ICCV*, 6201–6210.

Gao, J.; Chen, K.; and Nevatia, R. 2018. CTAP: Complementary Temporal Action Proposal Generation. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *ECCV*, volume 11206, 70–85.

Gao, J.; Shi, Z.; Wang, G.; Li, J.; Yuan, Y.; Ge, S.; and Zhou, X. 2020. Accurate Temporal Action Proposal Generation with Relation-Aware Pyramid Network. In *AAAI*, 10810–10817.

Gao, J.; Yang, Z.; Sun, C.; Chen, K.; and Nevatia, R. 2017. TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals. In *ICCV*, 3648–3656.

Gong, G.; Zheng, L.; and Mu, Y. 2020. Scale Matters: Temporal Scale Aggregation Network For Precise Action Localization In Untrimmed Videos. In *ICME*, 1–6.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970.

Jiang, Y.-G.; Liu, J.; Roshan Zamir, A.; Toderici, G.; Laptev, I.; Shah, M.; and Sukthankar, R. 2014. THUMOS Challenge: Action Recognition with a Large Number of Classes. http://crcv.ucf.edu/THUMOS14/. Accessed: 2020-June-1.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset. *CoRR*, abs/1705.06950.

Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; and Hager, G. D. 2017. Temporal Convolutional Networks for Action Segmentation and Detection. In *CVPR*, 1003–1012.

Li, Y.; Ji, B.; Shi, X.; Zhang, J.; Kang, B.; and Wang, L. 2020. TEA: Temporal Excitation and Aggregation for Action Recognition. In *CVPR*, 906–915.

Lin, J.; Gan, C.; and Han, S. 2019. TSM: Temporal Shift Module for Efficient Video Understanding. In *ICCV*, 7082–7092.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *ICCV*, 3888–3897.

Lin, T.; Zhao, X.; and Shou, Z. 2017. Single Shot Temporal Action Detection. In Liu, Q.; Lienhart, R.; Wang, H.; Chen, S. K.; Boll, S.; Chen, Y. P.; Friedland, G.; Li, J.; and Yan, S., eds., *ACM MM*, 988–996.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *ECCV*, volume 11208, 3–21.

Liu, Y.; Ma, L.; Zhang, Y.; Liu, W.; and Chang, S. 2019. Multi-Granularity Generator for Temporal Action Proposal. In *CVPR*, 3604–3613.

Liu, Z.; Luo, D.; Wang, Y.; Wang, L.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; and Lu, T. 2020. TEINet: Towards an Efficient Architecture for Video Recognition. In *AAAI*, 11669–11676.

Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; and Mei, T. 2019. Gaussian Temporal Awareness Networks for Action Localization. In *CVPR*, 344–353.

Qing, Z.; Su, H.; Gan, W.; Wang, D.; Wu, W.; Wang, X.; Qiao, Y.; Yan, J.; Gao, C.; and Sang, N. 2021. Temporal Context Aggregation Network for Temporal Action Proposal Refinement. In *CVPR*, 485–494.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks. In *ICCV*, 5534–5542.

Simonyan, K.; and Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *NIPS*, 568–576.

Su, H.; Gan, W.; Wu, W.; Qiao, Y.; and Yan, J. 2021. BSN++: Complementary Boundary Regressor with Scale-Balanced Relation Modeling for Temporal Action Proposal Generation. In *AAAI*, 2602–2610.

Tan, J.; Tang, J.; Wang, L.; and Wu, G. 2021. Relaxed Transformer Decoders for Direct Action Proposal Generation. In *ICCV*, 13526–13535.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*, 4489–4497.

Tran, D.; Ray, J.; Shou, Z.; Chang, S.; and Paluri, M. 2017. ConvNet Architecture Search for Spatiotemporal Feature Learning. *CoRR*, abs/1708.05038.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, 6450–6459.

Wang, L.; Tong, Z.; Ji, B.; and Wu, G. 2021. TDN: Temporal Difference Networks for Efficient Action Recognition. In *CVPR*, 1895–1904.

Wang, L.; Xiong, Y.; Lin, D.; and Gool, L. V. 2017. UntrimmedNets for Weakly Supervised Action Recognition and Detection. In *CVPR*, 6402–6411.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Gool, L. V. 2016. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *ECCV*, 20–36.

Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; and Cottrell, G. W. 2018. Understanding Convolution for Semantic Segmentation. In *WACV*, 1451–1460.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *ECCV*, volume 11219, 318–335.

Xiong, Y.; Wang, L.; Wang, Z.; Zhang, B.; Song, H.; Li, W.; Lin, D.; Qiao, Y.; Gool, L. V.; and Tang, X. 2016. CUHK & ETHZ & SIAT Submission to ActivityNet Challenge 2016. *CoRR*, abs/1608.00797.

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A. K.; and Ghanem, B. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *CVPR*, 10153–10162.

Yeung, S.; Russakovsky, O.; Mori, G.; and Fei-Fei, L. 2016. End-to-End Learning of Action Detection from Frame Glimpses in Videos. In *CVPR*, 2678–2687.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017a. Temporal Action Detection with Structured Segment Networks. In *ICCV*, 2933–2942.

Zhao, Y.; Zhang, B.; Wu, Z.; Yang, S.; Zhou, L.; Yan, S.; Wang, L.; Xiong, Y.; Lin, D.; Qiao, Y.; et al. 2017b. Cuhk & ethz & siat submission to activitynet challenge 2017. *arXiv preprint arXiv:1710.08011*, 8: 8.