RecapNet: Action Proposal Generation Mimicking Human Cognitive Process

Tian Wang, Yang Chen, Zhiwei Lin, Aichun Zhu, Yong Li, Hichem Snoussi, Hui Wang

Abstract

Generating action proposals in untrimmed videos is a challenging task, since video sequences usually contain lots of irrelevant contents and the duration of an action instance is arbitrary. The quality of action proposals is key to action detection performance. The previous methods mainly rely on sliding windows or anchor boxes to cover all ground-truth actions, but this is infeasible and computationally inefficient. To this end, this paper proposes RecapNet - a novel framework for generating action proposal, by mimicking the human cognitive process of understanding video content. Specifically, this RecapNet includes a residual causal convolution module to build a short memory of the past events, based on which the joint probability actionness density ranking mechanism is designed to retrieve the action proposals. The RecapNet can handle videos with arbitrary length and more importantly, a video sequence will need to be processed only in one single pass in order to generate all action proposals. The experiments show that, the proposed RecapNet outperforms the state-of-the-art under all metrics on the benchmark THUMOS14 and ActivityNet-1.3 datasets. The code is available publicly at https://github.com/tianwangbuaa/RecapNet.

***Index Terms*—Action Proposal, Action Detection, Residual Causal Convolution**

I. Introduction

Understanding human actions in video content is critical to building AI systems. Related areas include *action recognition* [1]–[9], *action detection* [10]–[17], *video segmentation* [18]–[21], *anomaly detection* [22]–[24] and so on. This paper mainly focuses on the action proposal generation in action detection. Given an untrimmed video, action detection aims to locate the action's temporal boundaries and at the same time to classify the detected action into correct categories. Since the open and long video stream inevitably contains irrelevant background noises, and actions may happen at any time with arbitrary durations, action detection is more challenging than action recognition.

This work is partially supported by the National Natural Science Foundation of China (61972016), the Fundamental Research Funds for the Central Universities (YWF-19-BJ-J-237), the Open Research Fund of Fujian Engineering Research Center of Public Service Big Data Mining and Application, Fuzhou, China, the European regional development fund-FEDER.

T. Wang and Y. Chen are with School of Automation Science and Electrical Engineering, Beihang University, China (email: wangtian@buaa.edu.cn (OR wangtian8704@gmail.com), chenyangwiz@buaa.edu.cn). Z. Lin, H. Wang are with School of Computing, Ulster University, United Kingdom (email: z.lin@ulster.ac.uk, h.wang@ulster.ac.uk). A. Zhu is with School of Computer Science and Technology, Nanjing Tech University, China (email: aichun.zhu@njtech.edu.cn). Y. Li is with School of Electronic Engineering, Beijing University of Posts and Telecommunications, China (email: yli@bupt.edu.cn). H. Snoussi is with Institute Charles Delaunay-LM2S FRE CNRS 2019, University of Technology of Troyes, France (email: hichem.snoussi@utt.fr).

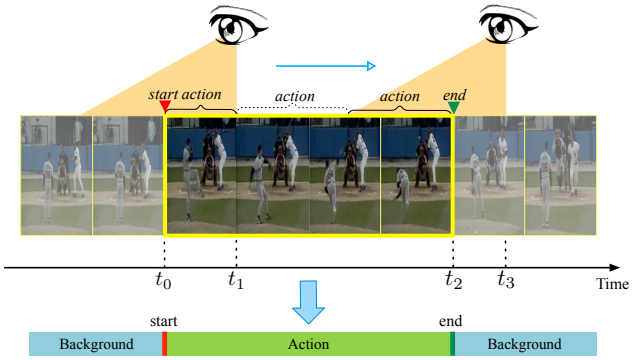

Fig. 1: The human cognitive procedure of understanding videos and retrieving action instances. A person focuses on deciding whether an action is *starting*, *ending* or *ongoing* at each time step by recapping the past events of a short period. A final decision is made based on the analysis of these local decisions. A new model (RecapNet) is introduced to mimic this procedure on how a neuron makes decisions based on a short memory of the past events. No sliding window or anchor box is required and the varying action duration problem gets well addressed.

In order to improve the action detection performance, the two-stage "proposal + classification" scheme has been widely studied [10]–[15], in which the first stage focuses on generating action proposals (*i.e.* locating temporal boundaries which cover the action instances well), and then the second stage is to classify the detected actions into correct categories.

In the research of action proposal generation, the key difficulty lies that the durations of the whole video sequences and the contained action instances are both arbitrary. SCNN [25] firstly implemented the multi-scale dense sliding window strategy (each video sequence is divided into dense short overlapping clips with different scales to cover the entire ground truth action) to address this problem. Since this is computationally expensive, *recurrent neural networks* have been designed to encode spatial-temporal features in the video stream instead [26], [27]. Other work, like TURN [28], TAL-Net [14] introduces the concept of "anchor" from object detection into action detection. TURN builds the proposal model by applying boundary regression to the proposals generated with multi-scale anchor boxes. TAL-Net extends the network's receptive field by utilizing dilated convolution layers in different dilation rates with anchor boxes. They all share the same issue that all the above models attempt to use predefined windows or anchor boxes with different lengths

to cover the ground truth actions. When the tIoU (temporal intersection over union) exceeds a certain threshold (usually 0.5), these eligible windows are then taken as the proposals. However, this is not the case for most of the video sequences. For example, in the benchmark dataset THUMOS14 [29], the action durations vary from a few seconds to more than 50 seconds, and as such flexible sliding window scales or anchor boxes are needed for improving recall. This commonly adopted design brings a mass of redundant computations and actually can not retrieve actions with extreme long durations. Different from all the above methods, TAG [30] proposes the concept of "actionness" to indicate whether an action happens at each time step. This design is enlightening but the TAG model's performance is relatively low due to lack of efficient actionness analysis strategy. In CTAP [15], a proposal actionness estimator on sliding windows is developed and an effective proposal ranking mechanism is designed to get the action proposals. However, this model still has the intrinsic drawback of the sliding window based methods. BSN [31] takes the action start-middle-end design from [32] to action proposal generation and achieves the state-of-the-art results. The critical problem of BSN is that BSN cannot handle video durations with various lengths flexibly, as it needs to interpolate the input features of the whole video into a fixed length. This coarse granularity design will inevitably lead to the loss of many useful spatial-temporal information. To address the above issues, this paper takes a different approach by simulating human cognitive process of video understanding.

As shown in Figure 1, the duration of the detected action happens between t_0 and t_2. However, the human does not realize this action until t_1, due to delay, and also does not realize that the action ends at t_2 until t_3. This delay is due to the fact that the human needs to receive enough information to make the final decision that the action starts from t_0 and ends at t_2. This decision process requires no sliding windows or anchor boxes and the decisions are not affected by the action durations. Inspired by this cognitive process, we propose a novel action proposal generation framework named RecapNet to mimic it. This RecapNet model recaps the events happened in a short period before the current time step. Moreover, the RecapNet outputs three scores for each time step, i.e. *starting score*, *ending score* and *actionness score*. The three scores represent whether an action is starting, ending and ongoing for each time step within the short period, respectively. The recapping is based on the memory of this short period. We design a residual causal convolution module to maintain this memory and learn semantics from it. After getting the recapping scores from each previous time step, we propose a joint probability actionness density ranking mechanism to retrieve the final action proposals in the global manner.

To sum up, the contributions of this paper include: (1) A novel recap mechanism is introduced to mimic the human cognition procedure of video sequences. (2) A residual causal convolution module is developed to model the contextual information in videos and to maintain a short memory of the past events. (3) A joint probability actionness density ranking mechanism is designed to form the global action proposal decisions. (4) The model can handle videos with arbitrary length and only needs to process the whole video in one single pass to get the action proposals. Thanks to the good model design and learning algorithms, our proposed method achieves state-of-the-art action proposal performance on the benchmark THUMOS14 [29] and ActivityNet-1.3 [33] dataset. The overall architecture of RecapNet is shown in Figure 2.

II. Releated Work

Object Recognition and Detection The success of object recognition with deep learning methods on ImageNet [34] triggered the bloom of deep learning research in recent years. A large body of investigations have been dealing with the task of object recognition and many elaborated models become the basic deep learning network architectures consisting of AlexNet [35], VGG [36], GoogLeNet [37], ResNet [38] and DenseNet [39], etc. As for the object detection, the deep models can be mainly divided into two-branches: one stage solutions including YOLO series [40]–[42], SSD [43], and two-stage solutions like R-CNN series [44], [45] and R-FCN [46]. The two-stage "proposal + classification" scheme from object detection is also transferred to the temporal action detection field and our action proposal model also follows this design.

Action Recognition Action recognition has attracted many researchers. Taking video segments with a fixed number of frames as input, action recognition needs to get them correctly classified into the corresponding categories. Earlier methods rely on handcrafted features including MBH [1], HOF [47], 3D-HOG [48], 3D-SIFT [49], iDT [2]. In the past few years, deep learning methods have achieved outstanding performances on action recognition, the most representative of which includes the 3D convolution network C3D [3], the pseudo 3D convolution network P3D [4], the two-stream network [5], I3D [6] that combines 3D convolution and the two-stream architecture, the RNN based network [7], CoViAR [50] that utilizs the motion vectors from videos, TSN [8] and the Non-local network [9].

Action Detection and Proposal Generation Action detection is to action recognition what object detection is to object recognition. In addition to classifying the action classes, action detection needs to locate the action instances in the temporal dimension. Most of the recent works apply the two-stage "proposal + classification" strategy [10]–[15], while there are also models adopting the one-stage scheme [16], [17], [51]. As for action proposal generation, SCNN [25] firstly put forward the multi-scale sliding window method. DAPs [26] and SST [27] utilize RNN to aggregate semantics in videos for alleviating the computation burden of the sliding window. TURN [28] introduces the multi-scale anchor boxes in proposal generation. TAL-Net [14] develops a set of dilation networks to expand the model's receptive field for better coverage of the action instances with various durations. TAG [30] introduces the concept of "actionness" and adopts watershed algorithm to group regions with high actionness scores as the proposals. CTAP [15] combines TAG and sliding window strategy. Faced with videos with arbitrary length, BSN [31] makes a compromise by interpolating the video features to a fixed stage

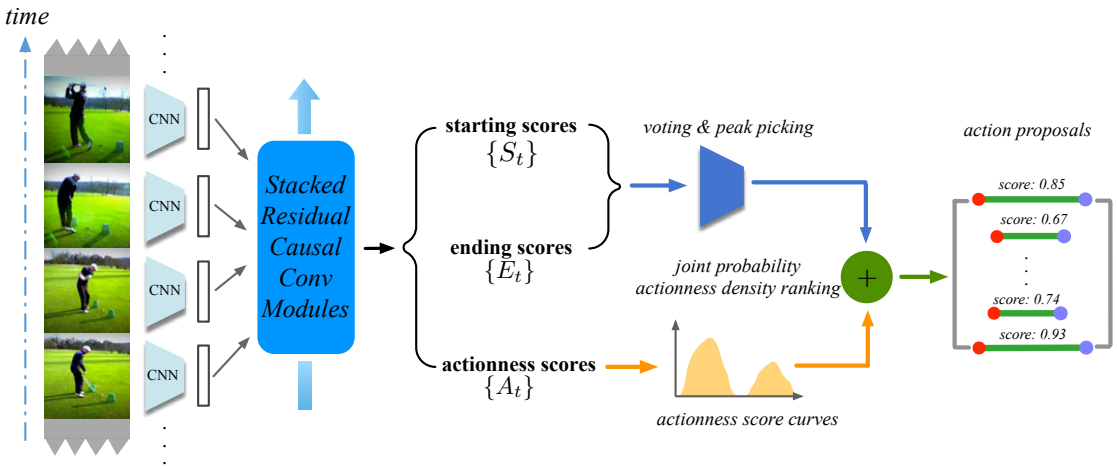

Fig. 2: The overall architecture of our proposed RecapNet which mimics the human recognitive procedure in understanding video content. The stacked residual causal convolution modules are developed to learn from a short period of the past events to obtain the *staring scores*, *ending scores* and *actionness scores* for previous time steps. Then we apply the voting and peak picking methods on the former two scores to get the candidate proposals. Finally, we use the joint probability actionness density ranking mechanism to obtain the action proposals. In this procedure, no sliding windows or anchor boxes are needed, and the varying action duration problem is well addressed.

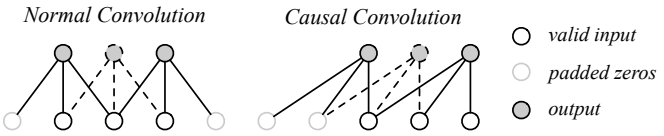

Fig. 3: The difference between normal convolution and causal convolution. Convolutions with kernel size 3 are depicted. With the front zero padding trick, causal convolution can be used for temporal reasoning.

and applies action start-middle-end design. In our work, we propose the RecapNet which mimics the human cognitive process in video understanding. It abandons the inefficient and complicated multi-scale sliding window or anchor box strategy and fundamentally solves the varying action duration problem in action proposal generation.

III. Our Approach

The goal of our RecapNet is to generate action proposals with high quality in untrimmed video sequences. The high quality can be reflected in the following two aspects: (1) The generated proposals should cover the ground truth well, which means the ground truth should be retrieved with high recall and has high overlap with the proposals. (2) The high recall and high overlap criteria should be met with a few number of proposals. In this way, there are fewer false predictions in the prediction results, and this in turn will bring less interference to the succeeding action recognition procedure.

A. Video Feature Encoding

Different from the object detection in still images, the semantic information encoded in the video context is critical to the action proposal generation. In our work, we choose the two-stream I3D network to perform the video feature extraction. The widely adopted two-stream network has made great success in action recognition. I3D takes a further step by extending the state-of-the-art image recognition network Inception-V1 [37] into the 3D form and then incorporates it into the two-stream architecture. Essentially, after pretraining on the largest action recognition dataset Kinetics [6], I3D becomes a good choice to extract the spatial-temporal features for the video sequences. I3D takes as input the stacked RGB frames and stacked optical flows, then outputs corresponding two-stream features.

Following the convention in state-of-the-art work [14], [28], [31], we first take apart each video into many non-overlapping units. Consider an input video sequence X with T frames in total, we divide X into $N = \frac{T}{\delta}$ non-overlapping units each of which has δ frames. Then we form the stacked RGB frames and optical flows of each unit as the input of the pretrained I3D network's spatial and temporal streams. In each stream, we choose the 1024D output of the last average pooling layer as the feature representation. Finally we concatenate them as the final 2048D feature vector set denoted as $F = \{f_i\}_{i=1}^{N} = \{f_{s,i}||f_{t,i}\}_{i=1}^{N}$, where $f_{s,i}$ and $f_{t,i}$ represent the output features from two streams separately with the i-th video unit as the input, and $||$ is the concatenation operation.

B. Residual Causal Convolution Module

Causal convolution, first put forward in WaveNet [52] to substitute RNN, gets state-of-the-art performance in text-to-speech audio generation. Its contribution is using stacked dilated convolution layers to achieve larger receptive field with fewer layers and fewer parameters, which successfully addresses RNN's hard to train and gradient vanishing problems.

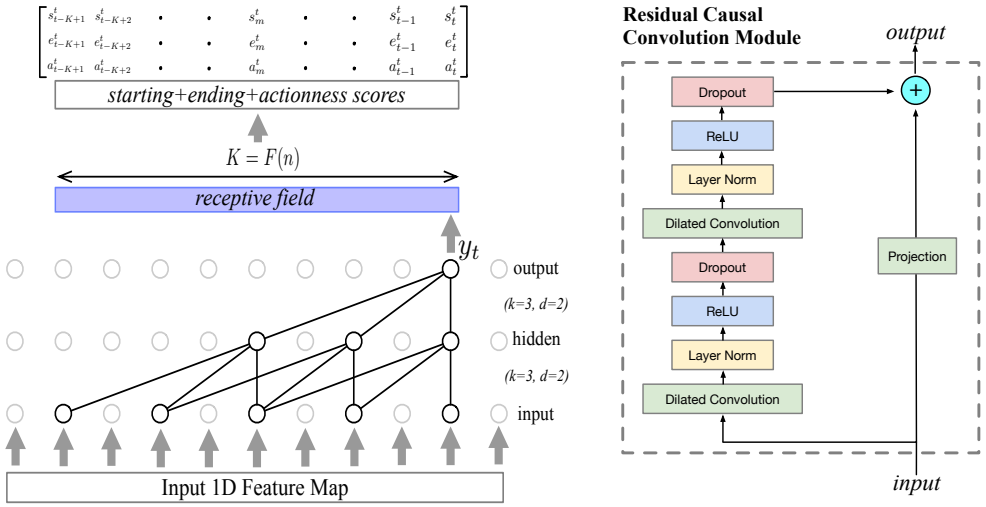

Fig. 4: **Left**: The illustration of multiple dilated convolution layers in one residual causal convolution module RCCM. $F(n)$ is the receptive field size of the top layer (here $F(0) = 1, F(1) = 5, F(2) = 9$). The output at each time step t contains *staring*, *ending* and *actionness* scores regarding to the past time steps in the receptive field. **Right**: The structure of one residual causal convolution module. We stack multiple such modules in our model.

"Causal" here means the network at time step t can only see inputs no later than t. Figure 3 shows the difference between normal convolution and causal convolution. The causal convolution is implemented by only padding zeros in front of the input to make the output and input have the same output size. In this manner, the output is only associated with inputs before it. Thus, causal convolution can be used for temporal reasoning.

Figure 4 depicts the residual causal convolution module RCCM in our RecapNet. A RCCM is made up of the dilated convolution layers, layer-normalization layers [53], dropout layers [54], ReLU layers [55] and the projection layer. For the multiple dilated convolution layers in RCCM, here we denote $F(i)$ as the i-th layer's receptive field size, which means how many time steps each neuron of this layer can see from the input layer $F(0)$. If we set the convolution stride to 1, then we have:

$$F(i) = \begin{cases} F(i-1) + (k_i - 1) \cdot d_i & i = 1, 2, \cdots \\ 1 & i = 0 \end{cases} \quad (1)$$

where k_i and d_i are the kernel size and dilation rate of the i-th convolution layer separately ($d_i = 1$ means no dilations). In our work, k and d is separately set to be the same for all layers in one RCCM with n convolution layers. In this way, the top layer's receptive field size can be calculated by $F(n) = F(0) + n(k-1)d$. After convolutions, the output of the top layer at time step t is only associated with $F(n)$ inputs at time interval $[t - F(n) + 1, t]$. Finally, we add the inputs to the module's outputs to form the residual connection, and the project layer is a 1×1 convolution that maps the input's channel to the output's channel. For simplicity, we denote $RCCM(n_c, n_n, k, d)$ as a RCCM with n_c convolution layers each with n_n neurons and the kernel size and dilation rate are set to k and d separately. For example, the receptive field size of the top layer of $RCCM(2, 512, 3, 2)$ is $F(2) = 1 + 2 \times (3-1) \times 2 = 9$.

In our work, we stack multiple RCCMs after the visual feature vectors in the RecapNet. Suppose the receptive field size of the last layer is K, then the output at time step t only receives the inputs from the past K time steps and maintains a short memory about what just happens in the past few seconds. By recapping the inputs within the receptive field, we let the neuron at the current time step t decide whether an action is *starting*, *ending* or *ongoing* at every past K time step. This is achieved by adding a 1×1 convolution layer with $3K$ kernels whose activation function is the sigmoid function to the end. In this manner, the output y_t at each time step t is a $3K$ dimension vector in the form of $\{[s^t_{t-K+1}, e^t_{t-K+1}, a^t_{t-K+1}], \cdots, [s^t_t, e^t_t, a^t_t]\}$, where $s^t_m, e^t_m, a^t_m (m \in [t-K+1, t])$ represent probabilities that the output neuron at time t predicts time m to be action *starting*, action *ending* and action *ongoing*.

C. Action Boundary Decision

The RCCMs make it that at time step t we get K decisions about the current action status from the future neurons between t and $t + K - 1$, i.e. $S_t = [s^t_t, s^{t+1}_t, \cdots, s^{t+K-1}_t]$, $E_t = [e^t_t, e^{t+1}_t, \cdots, e^{t+K-1}_t]$, $A_t = [a^t_t, a^{t+1}_t, \cdots, a^{t+K-1}_t]$, where S_t, E_t, A_t are the *starting*, *ending* and *actionness* score set, separately.

To get the action *starting* locations, we formulate two rules on boundary decision making: (1) voting scheme: there are more than V scores in S_t exceeding 0.5; (2) peak value picking: $\overline{S_{t-1}} < \overline{S_t} > \overline{S_{t+1}}$ ($\overline{S_t}$ is the average value of S_t). When any one of them is matched, we take the corresponding time step t as a possible action *starting* point. We collect all these *starting* points and form the candidate *starting* set C_s. Using same rules we can obtain the candidate *ending* set C_e. The voting scheme adopts decisions comprehensively from future neurons which can see the current time step. This

scheme can avoid missed detection due to misjudgments by a minority of the neurons. The peak value picking scheme pays extra attention to local maxima locations attracting the neurons, which may also be possible action boundaries.

For a *starting* time t_s^i from C_s and an *ending* time t_e^j from C_e, the interval $[t_s^i, t_e^j]$ becomes a candidate proposal if the condition $t_s^i < t_e^j$ gets satisfied. In this manner, we can get the candidate proposal set Φ_p containing all these intervals.

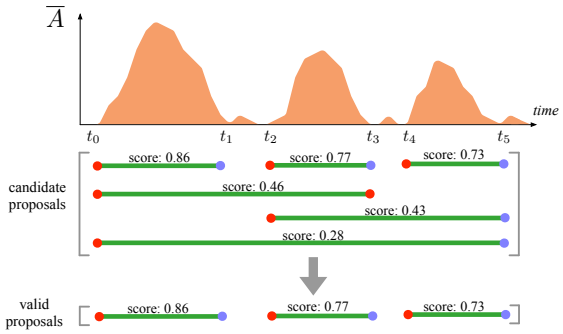

Fig. 5: The joint probability actionness density ranking mechanism. This mechanism rejects proposals which cannot cover the ground truth well.

D. Joint Probability Actionness Density Ranking

A perfect action proposal should satisfy that the temporal boundaries are precise and it has a high overlap with the target ground truth. Those proposals containing more than one action should be rejected as the average action information or actionness density is low. To this end, we propose the joint probability actionness density ranking mechanism to measure the effectiveness of a candidate action proposal, as shown in Figure 5. For a candidate action proposal $\phi_c = [t_s, t_e] \in \Phi_p$, its joint probability actionness density is defined as:

$$\begin{aligned} d_c &= \overline{S_{t_s}} \cdot \overline{E_{t_e}} \cdot \frac{\sum_{t_i=t_s}^{t_e} \overline{A_{t_i}}}{|t_e - t_s|} \\ \overline{A_{t_i}} &= \frac{\sum_{j=t_i}^{j=t_i+K-1} a_{t_i}^j}{K} \end{aligned} \tag{2}$$

Here $\frac{\sum_{t_i=t_s}^{t_e} \overline{A_{t_i}}}{|t_e - t_s|}$ is the actionness density metric representing the unit average actionness score of the candidate proposal, *i.e.* the curve area divided by the candidate proposal length in Figure 5. $\overline{S_{t_s}} \cdot \overline{E_{t_e}}$ denotes the confidence score of current proposal to have correct action boundaries. Thus we have the joint probability actionness density metric d_c to rank the candidate proposals. Higher this metric, more reliable the proposal.

E. Network training

Label Acquisition Given an action instance whose ground truth interval is $[u_i, u_j], 0 \leq i < j \leq N$, we define the following three regions: *starting* region $[u_{i-1}, u_{i+1}]$, *ending* region $[u_{j-1}, u_{j+1}]$ and the *action* region $[u_i, u_j]$. For unit u_l, which is in the receptive field of neuron n_t, if n_t sees u_l fall into any of these three regions, then we attach it with the corresponding label gs_l^t, ge_l^t, ga_l^t. For example, gs_l^t is assigned to 1 if it belongs to the start region, otherwise 0.

Unbalanced Sample Handling For the convenience of batch training, we randomly extract a segment with length L_w from each video sequence to form a batch of inputs. Then we have the binary label matrix in shape $[B, L_w, 3K]$ where B is the batch size. By counting the number of 1 and 0 values, we can get the statistical distribution of positive and negative samples (the ratio between positive and negative samples in both datasets is severely unbalanced). To overcome the problem of unbalanced training samples, we randomly choose negative/positive samples with the same number as the positive/negative ones, which is implemented by multiplying a binary mask matrix to the loss function. The 0 value in this mask matrix means that the corresponding samples are not involved in the back propagation.

Loss Function Our loss function consists of three parts:

$$L = L_{start} + L_{end} + \beta \cdot L_{action} \tag{3}$$

where L_{start}, L_{end} and L_{action} are the *starting* score loss, *ending* score loss and *actionness* score loss separately. β ($\beta \leq 1$) is a constant balancing the boundary loss and the actionness loss. Why β is needed is that the boundary scores and actionness scores have different statistical distribution and we found the actionness loss part converged faster in experiments. For any one of these three losses, we adopt the sigmoid cross-entropy loss function: (take L_{start} for example):

$$L_{start} = -\frac{1}{N_B} \sum_{i=1}^{N_B} (p_i \log q_i + (1 - p_i) \log(1 - q_i)) \tag{4}$$

where q_i is the predicted *starting* confidence score, p_i is the binary score label and N_B represents the number of video units used for training.

IV. Experiments

In the past few years, THUMOS14 and ActivityNet-1.3 became golden datasets for evaluating action proposals in deep learning based methods. In this section, We evaluate our RecapNet on THUMOS14 and ActivityNet-1.3 datasets respectively.

A. Datasets

THUMOS14 There are 200 and 212 untrimmed videos in the validation and test set [1], corresponding to 3007 and 3358 annotated action instances separately. Since the official training set is the UCF-101 [56] action recognition dataset which contains only trimmed videos but no temporal annotations, we train our model on the validation set following the convention. There are 20 action classes in THUMOS14 dataset.

ActivityNet-1.3 This dataset contains 19994 annotated YouTube videos in 200 classes and is divided into training, validation and test sets by 2:1:1. Since the annotation of the test set is not public available, we evaluate our model on the validation set.

[1] we exclude the falsely annotated video "270" during test.

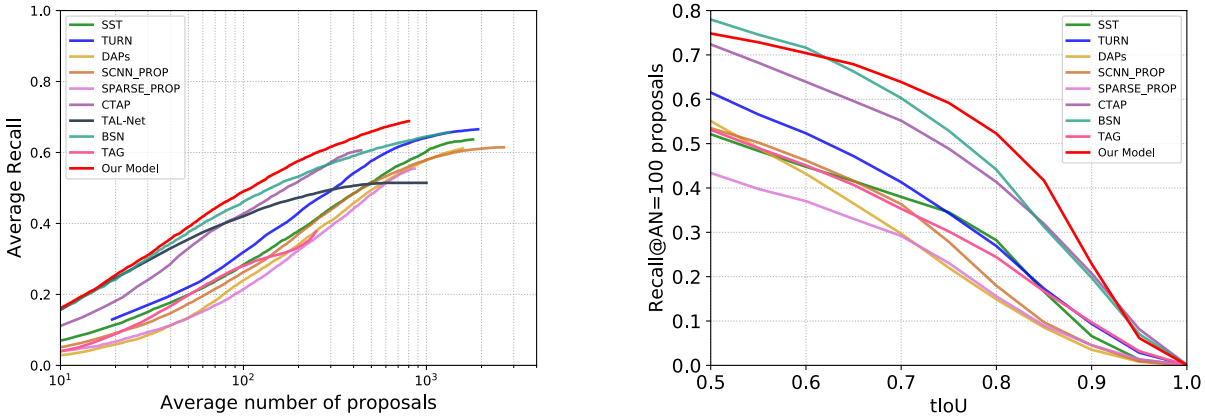

Fig. 6: Comparisons of our action proposal generation model with the state-of-art methods on THUMOS14 dataset in terms of the AR-AN metric and R@AN=100-tIoU. (**left**) Results under AR-AN metric. (**right**) Results under R@AN=100-tIoU.

We compare our model with the state-of-the-art methods on both datasets, and perform ablation studies on the THUMOS14 dataset, following the convention.

B. Experiment Setup

In our experiments, we apply the I3D model trained on the UCF-101 dataset to extract the unit-level video features with δ set to 16 on both datasets. We extract video segments with 150 unit length to form the training data ($L_w = 150$). We stack two residual causal convolution modules, which can be denoted as $RCCM(2,512,3,2) - RCCM(2,512,3,2)$, *i.e.*, each RCCM has two 512-kernel convolution layers whose kernel size is set to 3 and dilation rate is set to 2. In this manner, the receptive field size of the top layer is 17 which approximately lasts 9 seconds in the 30 FPS videos. In voting scheme of action boundary decision, we set V to 3 as the voting threshold. To prevent from overfitting during training, the dropout coefficient is 0.2 and we also add L2-norm multiplied with 1e-5 to the final loss function. β is empirically set to 0.2. We choose the Adam [57] as the optimizer with initial learning rate 3e-3, which is scaled down by a factor of 0.94 every 50 training epochs. Finally we apply the NMS with threshold 0.8 to get the action proposals. The optical flow is computed with TV-L1 algorithm [58]. We implement our model using the TensorFlow [59] framework.

C. Evaluation Metrics

In the action proposal task, the ground truth should be retrieved by generated proposals with high recall, which is often evaluated by the AR-AN metric. Given a certain Average Number of proposals (AN) per video, the Average Recall (AR) is calculated by averaging the recall across multiple tIoU thresholds. We follow the previous work by setting tIoU from 0.5 to 1.0 with step 0.05 on THUMOS14 dataset and 0.5 to 0.95 with step 0.05 on ActivityNet-1.3 dataset. On ActivityNet-1.3 dataset, the area under AR-AN curve (AUC) is also taken as another important metric, where AN varies from 0 to 100.

TABLE I: Comparison with state-of-the-art at some key points in terms of AR-AN on THUMOS14 dataset.

Method	@50	@100	@200	@300	@400	@500
SPARSE [60]	13.38	21.49	32.43	39.19	44.17	48.22
SCNN [25]	17.22	26.17	37.01	43.80	48.44	51.57
DAPs [26]	13.56	23.83	33.96	40.67	45.29	49.29
TAG [30]	19.81	28.04	33.34	-	-	-
SST [27]	19.90	28.36	37.90	44.27	48.75	51.58
TURN [28]	21.86	31.89	43.02	49.18	54.18	57.63
TAL-Net [14]	35.50	42.02	47.28	49.56	50.62	51.18
CTAP [15]	32.96	42.76	51.85	57.25	60.17	-
BSN [31]	37.46	46.06	53.21	56.82	59.05	60.64
RecapNet	**38.58**	**48.43**	**57.04**	**60.97**	**63.44**	**65.32**

TABLE II: Comparison with state-of-the-art at some key points in terms of R@AN=100-tIoU on THUMOS14 dataset.

Method	0.5	0.6	0.7	0.8	0.9
SPARSE [60]	43.39	37.03	29.19	15.56	4.57
SCNN [25]	53.49	46.20	36.36	17.96	4.51
DAPs [26]	55.10	43.20	29.70	14.88	3.53
SST [27]	52.13	44.78	37.98	28.20	6.60
TAG [30]	53.10	45.13	35.31	24.44	9.72
TURN [28]	61.53	52.33	41.31	26.90	9.37
CTAP [15]	72.37	63.93	55.16	41.36	20.74
BSN [31]	**77.99**	**71.62**	60.25	44.11	19.76
RecapNet	74.82	70.41	**63.89**	**52.27**	**22.89**

On the THUMOS14 dataset, the R@AN=100-tIoU curve is also chosen to report the localization precision of the proposals. The R@AN=100-tIoU curve shows the recall at different tIoU thresholds (from 0.5 to 1.0 with step 0.05) given a fixed Average Number (100) of proposals for each video. This metric indicates whether the proposals can cover the ground truth with high overlap.

D. State-of-the-art Comparisons

(1)**THUMOS14** We plot the AR-AN and R@AN=100-tIoU [2] curves in Figure 6 and list the results at some key points in

[2]We only have TAL-Net's results under the AR-AN criterion since there is no open data of the R@AN=100-tIoU results.

TABLE III: Comparison with state-of-the-art on ActivityNet-1.3 validation set under the AR@AN=100 and AUC criteria.

Method	TURN [28]	TAG [30]	MSRA [61]	SSAD [62]	CTAP [15]	BSN [31]	RecapNet
AR@AN=100 (val)	49.73	63.52	-	73.01	73.17	74.16	**75.62**
AUC (val)	54.16	53.02	63.12	64.40	65.72	66.17	**69.13**

TABLE IV: Comparison with state-of-the-art methods using C3D features on THUMOS14 dataset. The key points of AR-AN curve are reported.

Feature	Method	50	100	200	500	1000
C3D	TURN [28]	19.63	27.96	38.34	53.52	60.75
C3D	BSN [31]	**29.58**	37.38	45.55	54.67	59.48
C3D	RecapNet	29.44	**39.61**	**48.27**	**56.69**	**62.52**

TABLE V: Model design justification experiments of RNN *vs.* Causal Convolution in terms of AR-AN on THUMOS14 dataset.

	@50	@100	@200	@300	@400	@500
GRU	35.13	44.29	52.51	56.17	57.66	-
Causal Convolution	**38.58**	**48.43**	**57.04**	**60.97**	**63.44**	**65.32**

Table I and Table II. It can be observed that our RecapNet outperforms the state-of-the-art by a large margin on both curves. Especially, in AR-AN curve, our model significantly improves the average recall at AN=500 from 60.64% (BSN) to 65.32% by 4.68%. In R@AN=100-tIoU curve, our model has excellent performance in high tIoU regions and our model is the only model whose average recall exceeds 0.5 at the tIoU=0.8 point. This two metrics indicate that not only can RecapNet retrieve the ground truth with higher recall, but also the proposals generated by our model can cover the ground truth better with higher overlap.

(2)**ActivityNet-1.3** We report the result comparison on ActivityNet-1.3 dataset in Table III. On the validation set, our RecapNet outperforms all the state-of-the-art methods on AR@AN=100 and AUC criteria.

TABLE VI: Influence of Receptive field size and comparisons under AR-AN at some key points on THUMOS14 dataset.

Receptive Field Size (K)	@50	@100	@200	@300	@400
13	36.30	**48.50**	**57.90**	60.93	63.04
17	38.58	48.43	57.04	**60.97**	**63.44**
25	**38.99**	47.58	56.87	60.27	63.32

TABLE VII: Experiments to verify the contributions of voting and peak value picking rules in action boundary decision making. Results of AR-AN key points on THUMOS14 dataset are reported.

Voting	Peak	@50	@100	@200	@300	@400	@500
✓		38.46	48.38	56.84	60.82	63.29	65.05
	✓	33.64	40.18	44.89	-	-	-
✓	✓	**38.58**	**48.43**	**57.04**	**60.97**	**63.44**	**65.32**

TABLE VIII: Influence of voting threshold V and comparisons under AR-AN at some key points on THUMOS14 dataset.

V	@50	@100	@200	@300	@400	@500
3	**38.58**	**48.43**	57.04	60.97	63.44	**65.32**
5	37.42	48.05	**57.10**	**61.02**	**63.45**	65.21
7	37.11	47.74	56.58	60.47	62.58	-
9	36.53	46.97	55.72	59.02	-	-

E. Ablation Studies on THUMOS14

Following the convention, we perform ablation studies on the THUMOS14 dataset.

Experiments on Visual Features In order to prove that the model design of RecapNet rather than the visual features is the main reason for its excellent performance, we carry on a controlled experiment by substituting I3D features with C3D features. The C3D is pre-trained on the UCF-101 dataset, which is the same as the setting of BSN [31], and we take the outputs of the last but two fully-connected layer as the visual features. From the result in Table IV, C3D-RecapNet still outperforms the state-of-the-art C3D-Turn and C3D-BSN, which verifies the good model design of our RecapNet.

Evaluation of Residual Causal Convolution Modules As mentioned in Section III-B, the causal convolution design is better than RNN. To verify the effectiveness of this design, we replace the stacked causal convolution modules with two stacked GRU modules. To keep the consistency, each GRU module is set to have 512 neurons and other hyper-parameters during training remain the same. We choose GRU as GRU based model has much better performance than LSTM in our experiments. The results under AR-AN metric are shown in Table V. We can see that the causal convolution design indeed is a good substitute for RNN as it has much better performance.

Influence of Receptive Field Size To assess the influence of receptive field size K on the performance, we add additional experiments to track it. We change the first residual causal convolution module's dilation rate to 1 to set K to 13, *i.e.* $RCCM(2, 512, 3, 1) - RCCM(2, 512, 3, 2)$, and modify the second module's kernel size to 5 to set K to 25, *i.e.* $RCCM(2, 512, 3, 2) - RCCM(2, 512, 5, 2)$. We report the results of AR-AN key points in Table VI. It can be observed that the receptive field size has slight impact on the model's performance. This result is in line with our expectation as our model focuses on the local areas and then forms the global decisions. In acceptable volatility range, the performance of our model does not get affected when the size of local area changes, which proves our design's effectiveness.

Study of the Boundary Decision Making Rules To verify the contributions of the voting and peak value picking rules in action boundary decision making, we perform controlled experiments. Table VII shows that the voting scheme con-

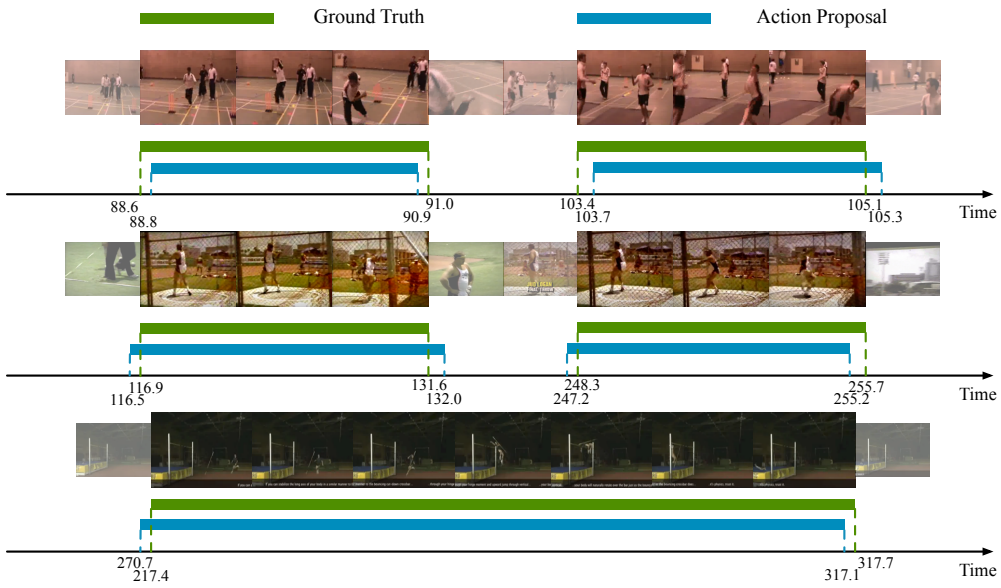

Fig. 7: Qualitative examples of generated action proposals on THUMOS14 dataset. It can be observed that (1) the detected boudaries are precise; (2) our model can handle actions with various durations.

TABLE IX: The contingency table of two model instances on the THUMOS14 dataset. Results under AN=100 and tIoU=0.5 are reported.

Model 1	Model 2	
	detected proposals	missed proposals
detected proposals	2357	134
missed proposals	127	711
statistics	χ^2=0.138, p-$value$=0.710	

TABLE X: The contingency table of two model instances on the ActivityNet-1.3 dataset. Results under AN=100 and tIoU=0.5 are reported.

Model 1	Model 2	
	detected proposals	missed proposals
detected proposals	6686	317
missed proposals	298	353
statistics	χ^2=0.527, p-$value$=0.468	

tributes most to the proposals. The peak value picking scheme pays extra attention to local maxima where neurons are excited and slightly improves the overall performance. The experiment results are exactly in consistent with our design intention. Note that the results with only voting scheme have already outperformed all the state-of-the-art. For the threshold V in voting scheme, we also perform the controlled experiments. As shown in Table VIII, there is no obvious performance difference between $V = 3$ and $V = 5$. But too big V may cause lower recall and fewer proposal numbers. This is reasonable as bigger V means more strict constraint.

F. Statistical Analysis

To evaluate the stability of our RecapNet's performance, we perform the McNemar's test on both the THUMOS14 and ActivityNet-1.3 datasets following [63]. For each dataset, we trained two model instances of our RecapNet on the training set until they are converged. Then we evaluated these two models on the test set with AN=100 and tIoU=0.5.

The McNemar's test is to check if the disagreements between two models match, $i.e.$, the number of proposals model 1 detects but model 2 misses should be equal to the number of proposals model 2 detects but model 1 misses. Thus the Hypothesis 0 (H_0) is that the two models have similar proportions of errors on the test set and the Hypothesis 1 (H_1) is that the two models have different proportions of errors on the test set. We report the contingency tables on both datasets on Table IX and Tabel X. On the THUMOS14 dataset, the McNemar's test statistic χ^2 with 1 degree of freedom and the corresponding p-$value$ are:

$$\chi_1^2 = \frac{(|134 - 127| - 1)^2}{134 + 127} = 0.138,$$
$$p\text{-}value_1 = 0.710.$$

On the ActivityNet-1.3 dataset, the statistic and the p-value are:

$$\chi_2^2 = \frac{(|317 - 298| - 1)^2}{317 + 298} = 0.527,$$
$$p\text{-}value_2 = 0.468.$$

Given the significance level $\alpha = 0.01$, since $p\text{-}value_1 > \alpha$ and $p\text{-}value_2 > \alpha$, we accept H_0 and reject H_1. This indicates that there is no difference in the disagreement between two model instances on both datasets and the performance our RecapNet of is stable.

To demonstrate the superiority of our RecapNet in terms of the statistical analysis, we report the McNemar's test statistic values χ^2 of the state-of-the-art methods in Table XI. It shows that our RecapNet achieves the lowest χ^2 on both the

TABLE XI: Comparison with state-of-the-art on THUMOS14 and ActivityNet-1.3 dataset in terms of the statistical analysis.

χ^2	SPARSE [60]	SCNN [25]	DAPs [26]	TAG [30]	SST [27]	TRUN [28]	CTAP [15]	BSN [31]	RecapNet
THUMOS14	1.136	0.582	0.985	0.296	0.213	0.238	0.170	0.312	**0.138**
ActivityNet-1.3	–	–	–	1.618	–	1.165	0.531	0.639	**0.527**

TABLE XII: Comparison of different proposal generation methods with the same action classifier (SCNN) on THUMOS14 test set. Results under mAP criterion with tIoU ranging from 0.3 to 0.7 are reported.

Method	0.7	0.6	0.5	0.4	0.3
SCNN [25]	5.3	10.3	19.0	28.7	36.3
SST [27]	-	-	23.0	-	-
TURN [28]	7.7	14.6	25.6	33.2	44.1
CTAP [15]	-	-	29.9	-	-
BSN [31]	15.0	22.4	29.4	36.6	43.1
RecapNet	**17.1**	**25.2**	**33.7**	**40.4**	**44.1**

THUMOS14 and ActivityNet1.3 datasets. This indicates that our RecapNet has the most stable performance.

G. RecapNet for Action Detection

To further demonstrate the effectiveness of our RecapNet, we follow the "proposal+classification" scheme that we feed the generated proposals to the SCNN [25] classifier and compare with other state-of-the-art models under the same classifier. In this task, the average Mean Precision (mAP) metric is used to evaluate the detection results. Table XII reports the result comparisons under mAP metric with tIoU ranging from 0.5 to 0.7. It's evident that RecapNet outperforms the state-of-the-art across all tIoUs on mAP. Most importantly, RecapNet is the first model achieves mAP@tIoU=0.5 exceeding 30%.

H. Speed and Efficiency

We evaluate our model's run-time performance both on the theoretical computational complexity and the real running time. Table XIII reports the theoretical complexity of our model. We deploy our model on the platform with one single Nvidia Titan XP GPU and an Intel i7-6850K CPU and measure the running speed with the FPS (Frames Per Second) metric. The file I/O, image processing, optical flow extraction and NMS operation are all performed on the CPU, while only the inference of the neural network is executed on the GPU. The speed of the whole procedure is 69.8 FPS. The fast speed gives the possibility of real-time action detection, which will be the future trend. Note that the state-of-the-art methods [14], [15], [31] did not offer the run-time performance, we cannot make comparisons.

I. Generalization of proposals

A good action proposal generation method should have good generalization ability, which means the model can also generate proposals with high quality on unseen action classes.

TABLE XIII: Theoretical computational complexity of our model. For a 16-frame video unit, the complexity is measured by number of parameters and FLOPs (Floating Point Operations).

	Params	FLOPs
I3D (Single Stream)	10.5 M	27.0 G
RecapNet	6.8 M	7e-3 G
Total	27.8 M	54.0 G

TABLE XIV: Generalization ability evaluation and comparisons with the state-of-the-art on the ActivityNet-1.3 validation set.

	AR@AN=100		AUC		χ^2
	seen(val)	unseen(val)	seen(val)	unseen(val)	
TAG [30]	68.10	66.40	-	-	0.875
CTAP [15]	74.06	72.51	66.01	64.92	0.700
RecapNet	75.38	**73.79**	69.54	**68.21**	**0.485**
	AR@AN=100		AUC		χ^2
	seen(val)	unseen(val)	seen(val)	unseen(val)	
BSN [31]	72.42	71.31	64.02	63.38	0.482
RecapNet	73.29	**72.36**	67.01	**64.21**	**0.346**

In TAG [30] and CTAP [30], the model is trained on *seen* 100 classes (the overlapped part between ActivityNet-1.2 and ActivityNet-1.3) and then directly evaluated on the other 100 *unseen* classes of ActivityNet-1.3. In BSN [31], the model is trained on *seen* 87 classes (a subset of ActivityNet-1.3 containing sports, exercise, and recreation actions) while tested on *unseen* 38 classes (a subset of ActivityNet-1.3 containing socializing, relaxing, and leisure actions). Following these settings, we take the absolute performance on *unseen* classes and the McNemar's statistic χ^2 as the metrics and conduct comparasions with the state-of-the-art methods. As shown in Table XIV, the performance of RecapNet on *unseen* subset only drops slightly and RecapNet achieves the best AR@AN=100 and AUC performance on the unseen data, which proves the generalization ability of the proposed RecapNet. Moreover, RecapNet has the most stable performance as it achieves the lowest χ^2 statistic value.

J. Qualitative Analysis

In this section, we select some output proposal samples of our model for qualitative analysis, as depicted in Figure 7. The ground truth action and the nearest top-ranked proposals are represented in parallel for a better demonstration. It can be observed that the generated proposals of our RecapNet are flexible, as actions with various durations (from about 1s to about 50s) get retrieved successfully. This excellent performance is due to our design that abandons the sliding window or anchor box strategy but mimics human cognitive procedure, which brings capability of handling extreme temporal variations.

V. Conclusion

This paper proposes a novel framework named RecapNet. For generating temporal action proposals, the model mimics the human cognitive process of understanding video content and it requires no sliding windows or anchor boxes. The proposed RecapNet is evaluated with the benchmark THUMOS14 and ActivityNet-1.3 datasets and it achieves the state-of-the-art results. In future, we plan to dig deeper into the cognitive process. For example, incorporating the attention mechanism [64] is of great value.

References

[1] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006, pp. 428–441.

[2] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3551–3558.

[3] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[4] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5534–5542.

[5] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 568–576.

[6] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[7] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4694–4702.

[8] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 20–36.

[9] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7794–7803.

[10] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," vol. 2, 2017.

[11] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 5727–5736.

[12] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," *arXiv preprint arXiv:1705.01180*, 2017.

[13] H. Xu, A. Das, and K. Saenko, "R-c3d: region convolutional 3d network for temporal activity detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5794–5803.

[14] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the Faster R-CNN Architecture for Temporal Action Localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1130–1139.

[15] J. Gao, K. Chen, and R. Nevatia, "CTAP: Complementary temporal action proposal generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 1–16.

[16] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017, pp. 1–12.

[17] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1417–1426.

[18] Y. Liu, F. Zhou, W. Liu, F. De la Torre, and Y. Liu, "Unsupervised summarization of rushes videos," in *Proceedings of the 18th ACM international conference on Multimedia (ACMMM)*. ACM, 2010, pp. 751–754.

[19] F. Zhou, F. De la Torre, and J. F. Cohn, "Unsupervised discovery of facial events," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 2574–2581.

[20] F. Zhou, F. De la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 582–596, 2013.

[21] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork convnets for video semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 852–868.

[22] T. Wang, M. Qiao, Z. Lin, C. Li, H. Snoussi, Z. Liu, and C. Choi, "Generative neural networks for anomaly detection in crowded scenes," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1390–1399, 2018.

[23] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, 2014.

[24] T. Wang, Z. Miao, Y. Chen, Y. Zhou, G. Shan, and H. Snoussi, "Aed-net: An abnormal event detection network," *arXiv preprint arXiv:1903.11891*, 2019.

[25] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1049–1058.

[26] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Daps: Deep action proposals for action understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 768–784.

[27] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "SST: Single-Stream Temporal Action Proposals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6373–6382.

[28] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[29] 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," http://crcv.ucf.edu/THUMOS14/, 2014.

[30] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang, "A pursuit of temporal accuracy in general activity detection," *arXiv preprint arXiv:1703.02716*, 2017.

[31] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[32] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, "Temporal action localization by structured maximal sums," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3684–3692.

[33] 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 (CVPR)*, 2015, pp. 961–970.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014, pp. 1–14.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[41] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 7263–7271.

[42] Redmon, Joseph and Farhadi, Ali, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[44] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1440–1448.

[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 91–99.

[46] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 379–387.

[47] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[48] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2008, pp. 275:1–10.

[49] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the ACM on Multimedia Conference (ACMMM)*, 2007, pp. 357–360.

[50] C.-Y. Wu, M. Zaheerc, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Compressed Video Action Recognition." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6026–6035.

[51] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the ACM on Multimedia Conference (ACMMM)*, 2017, pp. 988–996.

[52] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proceedings of the ISCA Speech Synthesis Workshop (SSW)*, 2016, pp. 1–15.

[53] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.

[56] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[58] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L 1 optical flow," in *Proceedings of the Joint Pattern Recognition Symposium*, 2007, pp. 214–223.

[59] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *Proceedigns of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, vol. 16, 2016, pp. 265–283.

[60] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1914–1923.

[61] T. Yao, Y. Li, Z. Qiu, F. Long, Y. Pan, D. Li, and T. Mei, "Msr asia msm at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and densecaptioning events in videos," in *CVPR ActivityNet Challenge Workshop*, 2017.

[62] T. Lin, X. Zhao, and Z. Shou, "Temporal convolution based action proposal: Submission to activitynet 2017," *arXiv preprint arXiv:1707.06750*, 2017.

[63] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[64] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.