Detecting Temporal Proposal for Action Localization with Tree-structured Search Policy

Xinyang Jiang
College of Computer Science and Technology, Zhejiang University
xinyangj@zju.edu.cn

Siliang Tang*
College of Computer Science and Technology, Zhejiang University
siliang@zju.edu.cn

Yang Yang
College of Computer Science and Technology, Zhejiang University
yangya@zju.edu.cn

Zhou Zhao
College of Computer Science and Technology, Zhejiang University
zhaozhou@zju.edu.cn

Yin Zhang
College of Computer Science and Technology, Zhejiang University
zhangyin98@zju.edu.cn

Fei Wu
College of Computer Science and Technology, Zhejiang University
wufei@cs.zju.edu.cn

Yueting Zhuang
College of Computer Science and Technology, Zhejiang University
yzhuang@zju.edu.cn

ABSTRACT

Understanding the semantics in videos is a complex but crucial task in video analysis. This paper focuses on localizing category-independent events, actions or other semantics in an untrimmed video, referred as salient temporal proposal localization. Traditional methods like sliding window have a high computational cost due to the densely sampling of different video segments. We propose a reinforcement learning based method, which trains a localizer that learns a search policy that, instead of exploring every video segment, finds an optimal search path to locate a salient proposal based on the currently observing video segment in a tree structure, therefore reduces the number of video segments fed into the proposal detector. In each search step, a localizer is trained to iteratively select the next sub-region containing salient proposals to continue the search, and a proposal detector is trained to recognize salient proposal from the sub-regions. The experiments demonstrate that our method is able to precisely detect salient proposals with a comparable recall and with much fewer candidate windows.

CCS CONCEPTS

•Theory of computation → Reinforcement learning; •Computing methodologies → Activity recognition and understanding;

KEYWORDS

Temporal Localization; Salient Proposals; Reinforcement Learning

*Siliang Tang is the correspondence author.

MM'17, October 23–27, 2017, Mountain View, CA, USA.

DOI: https://doi.org/10.1145/3123266.3123362

ACM Reference format:
Xinyang Jiang, Siliang Tang, Yang Yang, Zhou Zhao, Yin Zhang, Fei Wu, and Yueting Zhuang. 2017. Detecting Temporal Proposal for Action Localization with Tree-structured Search Policy. In *Proceedings of MM'17, October 23–27, 2017, Mountain View, CA, USA.*, , 9 pages.
DOI: https://doi.org/10.1145/3123266.3123362

1 INTRODUCTION

Understanding semantics in the videos is a complex but crucial task in video analysis. Although, in recent years, impressive progress has been made in video understanding, such as action recognition [16], action localization [15, 17], multimedia event detection [21], video understanding still remains an extremely challenging research problem. A large portion of the videos in the modern internet era are produced by people under unconstrained conditions. In real applications, these untrimmed videos are highly unconstrained in space and time, and complicated semantics can scatter around a video, so the detector needs to not only recognize the content of the video, but also to localize where certain semantics or event appears on the video time-line. We call this task action localization, where in this paper, action refers to any types of semantics or events happen in videos. Action localization has become a hot research area and can potentially save time and computational cost for application like video surveillance.

In this paper, we focus on an important technique to lower the computational cost for action localization, which trains a detector to localize a salient proposal of a category-independent event or action in a video. We call it salient temporal proposal detection in this paper.

Most of the state-of-art video localization methods localize the events or actions by densely sampling the segments of videos and run each segment through action classifiers. These segments can be defined as single video frames [23], or a temporal windows containing multiple frames to retain the temporal information among frames [7, 15]. Work like [15] achieves state-of-art performance by densely sampling windows with different time length as shown

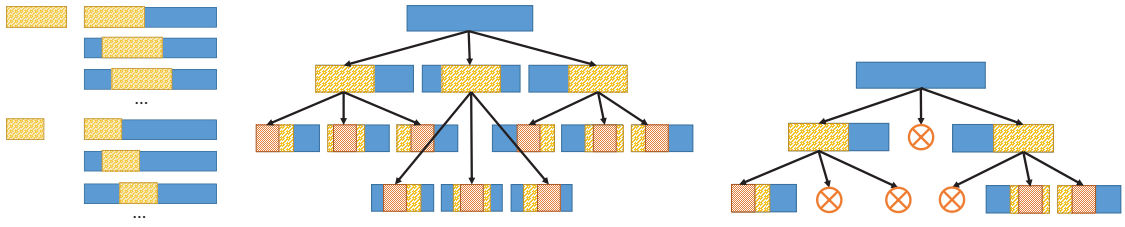

Figure 1: The illustration of three different search policies for proposal localization. Sliding Window (a) approach enumerates window with different sizes at different positions. The pyramid approach (b) forms a tree structure, the child node enumerates sub-regions within the region of their father node. (c) uses a search policy to prune the branches of the tree structure.

in Figure 1 (a), and feeding the whole sequence of frames into the action classifier. This type of approach is called the sliding-window approach.

One of the shortcomings of locating actions from the densely sampled video segments is the computational cost. In order to achieve good performance, the classifier could be a complicated multi-category classifier like a very deep neural network, and it requires all kinds of time consuming video features such as improved Dense Trajectory [20] and Fisher Vector [13] as input. The feature extraction and classification of all the densely sampled video segments will cost large amount of computation resource. Salient temporal proposal detection is an effective way to decrease the number of input segments for the video localization. It trains a salient proposal detector that rules out the background segments, and the classifiers are only ran on detected temporal regions that potentially contain important semantics. In this paper, we refer these detected temporal regions as the salient temporal proposals. As far as we know, despite commonly used in image object detection [5], salient proposal detection for videos has not made much progress. Works like action tube [6] and tubelets [8] propose novel approaches to locate a person and his/her actions in a video, but not for general video semantics. Our research focus on a more general localization problem, including any kinds of events, actions or other salient semantics in videos. Following the idea of faster-RCNN [14] in image object detection, [15] proposes a proposal network to detect salient proposals from candidate windows generated by the sliding window approach.

Although the proposal detector reduces the computational cost for action classifier by ruling out the background temporal regions, the proposal detection itself still needs to explore all the densely sampled video segments and search for the salient proposals with brutal force. As a result, we hope to find a new search policy which adjusts the search set of video segments based on what has been explored, so that the detector does not need to run on every segment like sliding window. In this paper, we propose learning this search policy with reinforcement learning. We train a proposal detector that can find an optimal search path to locate a salient proposal based on the current video segments it observes, which decreases the total number of video segments the model explores. Different from sliding window, our method generates proposal candidates for the proposal detector by enumerating the temporal windows in a tree structure, as shown in Figure 1 (b). To reduce the total number of windows fed into the detector, a search policy is used, so that the search is only continued on the child nodes that potentially contain the salient proposals. As a result, the search tree is pruned and fewer tree nodes are explored in the video, as showed in Figure 1 (c).

In reinforcement learning, an agent is trained to interact with an environment and receives reward as feedback. The goal is to train the agent to get as much positive reward as possible from the environment. There has been a lot progress applying reinforcement learning in image object detection [1, 2, 11], where the object detector (agent) is trained to interact with the images (environment) to search for the true object (reward). The actions the object detector take in every step of interaction are usually to adjust the position and size of the window. A similar reinforcement learning framework can be applied for salient proposal localization in video, where the environment the proposal detector interacts with is a video and the action is adaptively changing the position and size of the temporal window.

Detectors trained by previous reinforcement learning methods are usually for category dependent detection, which means a new agent has to be trained for each category individually. As a result, with n categories, each of the n detectors has to run a search on the video in order to locate all the objects. To make both training and testing phase of the reinforcement learning more efficient, we propose training an agent that searches for category independent salient proposals. As a result, by separating the localization process with the event classification process, only one agent (a *localizer*) has to be trained. As far as we know, there has not been a reinforcement learning based method that is specifically designed for category-independent video salient proposal.

Another challenge for reinforcement learning based method is to localize multiple proposals simultaneously in a video. In most of the previous works, detectors are trained to search one particular proposal in a single search. In order to locate all the proposals in the video, the detector has to run repeatedly until no proposal is left undetected. By applying a tree-structured recursive search, our method is able to locate all the salient proposals in one search. In each step of this recursive process, the detector selects all the sub-regions that contain the salient proposals and a new iteration of the search is initiated in every sub-region.

Overall, this paper proposes a reinforcement learning based methods that localizes the category-independent salient temporal proposals in video. By adapting reinforcement learning, we designed a tree-structured hierarchical search policy that makes the localization process more efficient and has the ability to locate multiple salient proposals in a single search. The rest of the paper is organized as follows: Section 2 introduces the related works to our method; Section 3 introduces the detailed formulation of our method; Section 4 reports the experiments result; and Section 5 is the conclusion.

2 RELATED WORK

2.1 Salient Region Proposal

Salient region proposal has been extensively researched and applied in image understanding. Unsupervised methods like selective search [19] detect the proposals by clustering the image segmentations with similar low-level features, which proves to have the ability to detect region proposals fast with a high recall rate. Traditional supervised models like BING [3] and Conditional Random Field [22] trains a proposal detector to predict the salient region based on the image's low-level features, which can be more efficient on the testing phase compared to the unsupervised models. For example, BING trains a Support Vector Machine to detect salient proposals based on Normed Gradient feature; the Conditional Random Field predicts a saliency map by modeling the correlation among the saliency labels for image patches and use SIFT descriptor as the input feature. Deep learning method like proposal network [4, 14] has shown significant performance improvement compared to traditional machine learning model with hand-craft low-level image features. Faster-RCNN directly feeds raw pixels within the sliding window into a CNN called the proposal network. To further accelerate the algorithm, R-FCN utilizes a full convolutional network. Without the fully connected layers, all the layers output can be reused for the windows in a single image and the number of model parameters also largely reduced.

Compared to image region proposal, fewer researches have been conducted in video temporal proposal. Research has been carried out to detect salient regions specifically for human actions. [8] proposes an unsupervised approach which uses segmentation methods to cluster similar voxel into a spatial-temporal action proposal called tublets. [6] uses CNN to detect action region in each frame and links the action region to form an action tube. [23] models the event detector as an attention LSTM. The densely sampled video frames are fed into the LSTM one by one, and the LSTM assigns an event label for each frame. [15] proposes a 3D proposal network to detect general temporal region proposals. Similar to faster-RCNN, the model feeds raw video frames within a sliding window into a 3D-CNN based proposal network.

2.2 Reinforcement Learning

Reinforcement learning has been a trending research topic in recent years. As the fast advance in deep learning, the performance of reinforcement learning rapidly advanced by integrating the deep network models, especially in robotics and artificial intelligence. Recently some works [1, 2, 11] explore the potential of reinforcement learning in the area of image object localization. The main idea of these works is similar, which is to treat the object detector as an agent and the image as the environment the agent interact with. The process of searching an object in an image is defined as a Markov Decision Process (MDP). The agent observes a series of regions in the image step by step and stops when it finds a region containing an object. The goal is to train the agent to find the object region both efficiently and accurately. The difference among these works lies in the action or reward configurations. For example, in [2] the agent transforms its current observed region in order to find a region that has a high IoU with the groundtruth object region. In [1] and [11], the agent chooses the next observed region by selecting one from a set of candidate regions. Rather than learning search policy based on a sliding window schema, [10] proposes to search for the object based on a tree structure, and uses the reinforcement learning to train the agent to localize the object from larger coarse windows to a smaller more precise bounding boxes.

[24] is the first work to use reinforcement learning in video action localization. In this work, each step of the MDP, the agent is trained to glimpse one frame in arbitrary position of the video. After finishing observing a certain proportion of the frames, the agent will emit a detection result. Since the agent observes single frames rather than a temporal region of the video, the context and temporal information in a temporal region could be disregarded. Instead of detecting category independent salient proposals, [24] is an end-to-end video action localization methods.

3 MODEL FORMULATION

Our goal is to take a long video clip and output a set of temporal region proposals that have the potential to contain video events or actions, so these proposals can be further used for action localization, event detection or other video applications. Our proposed method searches the salient temporal regions in a hierarchical fashion. Starting from larger sized windows, our algorithm step by step works its way down to the smaller scale sub-regions in the larger windows. There are two components in our model, i.e. the localizer and the detector. The localizer decides which of the sub-regions the model continues to search for the salient proposals. The detector decides if the current region should be emitted as the final salient proposal. At each step, the model first uses detector to determine if the current region is a salient region, and then uses localizer to select the sub-regions within the current region to continue the search. This pyramid configuration allows the model to adapt with long video clips and detect multiple salient temporal proposals simultaneously. Rather than locate one salient proposal in a single search, our model is able to locate all the regions in one run. That is because the localizer can locate all sub-regions that potentially contains salient proposals in the current search window at the same time, so that eventually all the salient proposals will be emitted. The search procedure of our model is demonstrated in Figure 2.

Essentially, the localizer generates a set of candidates by going through a search tree, and the detector selects the final result from the candidates. In this way, compared to directly feeding the detector with all the search windows in the pyramid configuration or with sliding window, the number of samples needed to be run by the proposal detector is reduced.

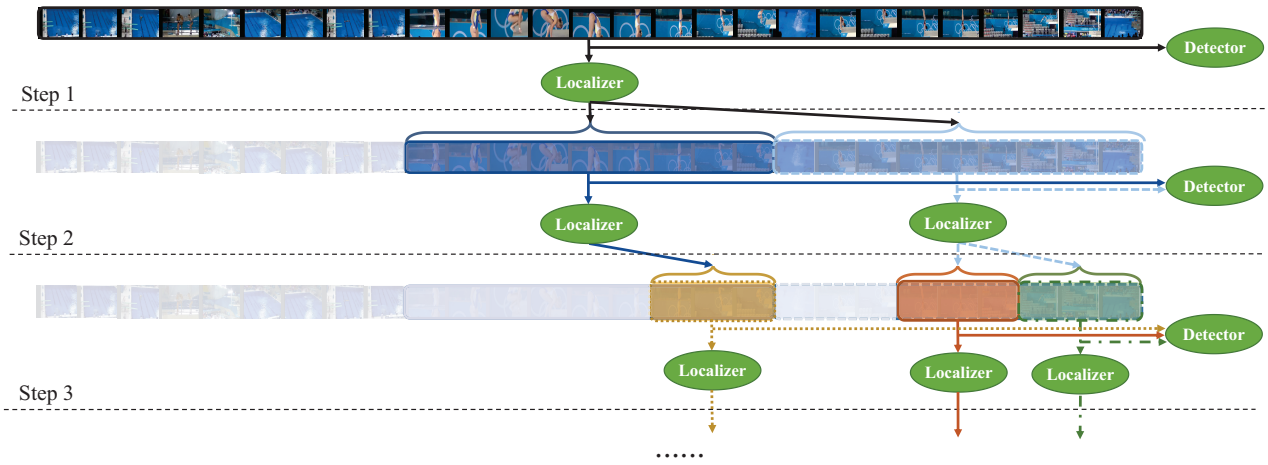

Figure 2: The illustration of the hierarchical search procedure of our proposed model. In step 1, the localizer selects the second and third sub-region to continue the search, which creates two search branches respectively. In step 2, for the first branch, the localizer selects the last sub-region and for the second branch, the localizer selects the first and second sub-region. In each step, the selected sub-regions are fed into a detector to decide if it should be emitted as the salient proposal.

3.1 Proposal Localizer

The localizer learns the search policy for the proposal detection. At each search step, it decides which of the sub-regions potentially contains the salient proposals and conducts further search in these regions.

3.1.1 The MDP process. The searching process of the localizer can be viewed as a Markov Decision Process. Given a video clip, the localizer sequentially observes temporal regions to search for salient proposals based on a hierarchical structure. Given a temporal region of a video, the next temporal region to observe is selected from one of the k sub-regions inside the currently observed regions that potentially contain the salient proposal. The agent receives a reward each time it performs an action and the goal is to maximize the total discounted reward during the entire running episode. The rewards the localizer receives reflect how accurate it covers the true salient proposals in its search path. Overall, the decision process of the agent can be formulated as an MDP, where the states are the different temporal regions the localizer currently observe, and the actions are the k sub-regions that can be chosen for the localizers and detector to observe next. The actions, states and rewards of our proposed MDP model are detailed as follows.

Action The actions in the proposed model are defined as selecting a new region for the localizer and the detector to observe next. In our model, the localizer selects one of the k pre-defined sub-regions within the current observed region that potentially contains the salient proposal. In the experiments, we define the sub-regions as three overlapped regions with the time length of 75% of the original temporal regions, which are the front 75 percent, middle 75 percent and last 75 percent of the time-line of the original regions. The localizer stops the search when the length of the observed region is less than a certain threshold.

State The state representation is the concatenation of a feature vector of the current observed temporal region and a vector of history actions. The feature vector of the observed region can be extracted by any available video feature descriptor. In our experiments, for any position t in the video, we extract a short clip of video with the length of 16 frames and use a 3D convolution neural network called C3D to pre-compute the feature of the short clip as the corresponding feature representation at position t. Given a temporal region starting at t_s and ending at t_e, it is represented as a feature vector $\mathbf{r}$ using max-pooling:

$$\mathbf{r} = \max_{t=t_s}^{t_e} C3D(t) \tag{1}$$

where function $C3D(t)$ extracts the C3D feature at position t. Therefore, for a currently observed state corresponding to a temporal region containing k subregions, the representation of the state denoted as $\mathbf{s}$ is the concatenation of the feature representation of its sub-regions r_k:

$$\mathbf{s} = (\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_k) \tag{2}$$

The history actions are defined as a binary vector to indicate which action is taken in the past. Each action is represented as a k-dimensional vector with the values of all dimensions being zeros except the one corresponding to the action taken.

Rewards The reward $r(\mathbf{s}, a)$ is a numerical value that the localizer receives each time it interacts with the video by performing an action a. The localizer learns a search policy to maximize the reward it receives from the video. We hope the localizer to learn a search policy that precisely locates a ground-truth region proposal at the end of each search path along the tree, and in the meantime, reduce the number of false positive proposals. Therefore, we need a reward function that encourages the localizer to move closer to a ground-truth proposal in each search step. Assuming the localizer at state $\mathbf{s}$ has an associated bounding window w. After taking

action a, the localizer transfers to state $\mathbf{s}'$ whose associated region is denoted as w'. The reward function is defined as follows:

$$r(\mathbf{s}, a) = \max_{1 \le i \le n} sign(IoU(w', g_i) - IoU(w, g_i)) \quad (3)$$

As we can see in Eq. 3, n is the number of the ground-truth salient proposals in the video and g_i is the i-th ground-truth window. $IoU(w, g)$ is a function that measures how accurate the current region w covers the ground truth region. It calculates the Intersection-over-Union(IoU) between current region w and the i-th ground truth region g_i. If the current region w has a lower IoU compared to the next region w', it means by taking action a, the localizer is moving toward a ground-truth salient proposal. Otherwise, it is moving away from a ground-truth salient proposal. Note that our method does not specify which ground-truth window the localizer should move towards in a certain step. As long as it moves towards one of the all ground-truth, it gets a positive reward +1. On the other hand, by taking action a, if even the largest IoU score of next window w' is still lower than current region w, it means this action does not help localizer move towards any of the ground-truth salient proposal, hence the localizer receives a negative reward −1. In this way, the localizer is allowed to localize any ground truth freely in one of its search paths, which is important for the localizer's ability to search for multiple salient proposals simultaneously.

3.1.2 Q-Learning. The optimal search policy of salient proposals is learned with reinforcement learning. Due to the high-dimensional continuous states and limited number of actions in our MDP formulations, the value-based reinforcement learning method called Q-Learning is the suitable model for our problem. Following [12], we use a deep neural network to predict the discounted total rewards given the current state $\mathbf{s}$ and action a, denoted as $Q(\mathbf{s}, a)$. Based on the Q-network, at each step, the agent will choose the action a that maximize $Q(\mathbf{s}, a)$ given the current state $\mathbf{s}$. As $Q(\mathbf{s}, a)$ is a function that approximates the total discounted rewards: $R(\mathbf{s}, a) = \sum_{t'=t}^{T} \gamma^{t'} r_{t'}$, where gamma is the discount rate for each step. The optimal $Q(\mathbf{s}, a)$ obeys the following Bellman Equation:

$$Q(\mathbf{s}, a) = r + \gamma \max_{a'} Q(\mathbf{s}', a') \quad (4)$$

where $\mathbf{s}'$ is the next state after the agent takes action a. As introduced in the former section, the state $\mathbf{s}$ is a concatenation of the feature representation of its k sub-regions. Since any of the k actions a_k correspond to a specific sub-region $\mathbf{r}_k$, the approximated discounted rewards of the action a_k depends only on $\mathbf{r}_k$. Therefore, we separate the original $Q - network$ into k isolated sub-networks with shared parameters, which further reduces the number of parameters and accelerates the training. For the k-th action with state s:

$$Q(\mathbf{s}, a_k) = Q'(\mathbf{r}_k, a_k) \quad (5)$$

where Q' is the new Q-network that shares the same parameter for different actions.

Based on the Bellman Equation, $Q(\mathbf{s}, a)$ can be trained by minimising the following loss function:

$$L(\theta) = \sum_{\mathbf{s}, a, \mathbf{s}'} (y - Q(\mathbf{s}, a; \theta))^2 \quad (6)$$

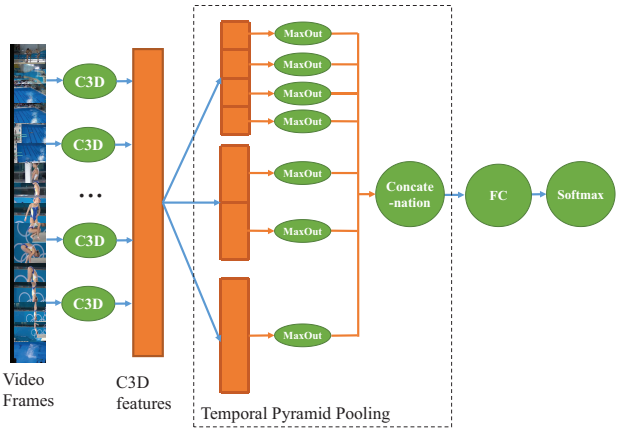

Figure 3: Illustration of the detector models. A sliding window is utilized to extract a sequence of C3D features. Then the C3D features are fed into a temporal pyramid pooling layer. In this example, the C3D sequence is split into sub-regions in three ways, i.e. 4, 2 and 1 equal-length sub-regions respectively. The features in sub-regions are maxed out and concatenated into a long feature vector as the input of the following two fully connected layers and softmax layer.

where θ are the parameters of the model and $y = r + \gamma \max_{a'} Q(\mathbf{s}', a'; \theta')$. θ' are the parameters from the last training iteration.

A stochastic gradient descent method is used to optimize the loss function. At the $i+1$-th iteration, the agent uses previously obtained θ_i to run one-step on a video, and obtains a training sample $(\mathbf{s}, a, y)$. Then by taking the derivatives of the loss function, θ_{i+1} is updated as follows:

$$\theta_{i+1} = \theta_i + \beta(y - Q(\mathbf{s}, a; \theta))\nabla_{\theta_{i+1}} Q(\mathbf{s}, a; \theta_i) \quad (7)$$

The agent obtains training samples $(\mathbf{s}, a, y)$ by consecutively running simulations on videos. As a result, if the samples are used to update the parameters in the same order, there will be very high correlation among the training samples, which leads to inefficient and unstable learning. Here we use a memory buffer to store the training samples, and at each iteration, a mini-batch of samples is sampled from the buffer to update the model. A ϵ-greedy policy is used for the agent to choose actions during training.

3.2 Proposal Detector

The proposal detector is responsible for generating the final result of the salient proposals. It observed every candidate the localizer generates in the search path and decides if it should be emitted as a salient proposal. We use a modified C3D [18] structure to build our network. First we apply a sliding window with a length of 16 frames on the input video region. Then, the frames in each window are fed into a C3D network. We took the output of the 4096-dimensional fully connected layer before the softmax layer. In order to capture the temporal information on different scales, a novel pooling layer called temporal pyramid pooling layer is applied. Pooling filters with different time length are applied on the

C3D's output. To create a fixed length output for the video regions with different length, the number of filters that applied on the video regions are fixed, while the length of the pooling filters varies based on the length of the video regions. To achieve this, we split a video region into sub-regions with equal length. Each of the sub-regions corresponds to a pooling filter, and the max-pooling is conducted on each of the sub-regions/filters. The split is conducted in different scales so that different lengths of filters are applied. Eventually the outputs of all the filters are concatenated to form an input fed into the following 2 fully connected layers and softmax layer. As a result, the length of the temporal pyramid pooling layer is always the number of the sub-regions multiplying 4096. Figure 3 shows the network structure of our model.

Algorithm 1 shows that given a video clip, how the localizer and detector collaboratively search for the salient proposals.

Algorithm 1 The search procedure of the proposed model

Input: Video clip $\mathbf{s}_0$;
Output: The set of salient proposals P
  function ProposalSearch($\mathbf{s}$)
    for each sub-region $\mathbf{r_k}$ in $\mathbf{s}$ **do**
      $score_1 = ProposalDetector(\mathbf{r_k})$
      $score_2 = Q'(\mathbf{r_k}, a_k)$
      if $score_1 > 0.5$ **then**
        Add position and size of $\mathbf{r}_k$ to P
      end if
      if $score_2 > 0$ **then** ProposalSearch($\mathbf{r_k}$)
      end if
    end for
  end function
  initial $P = \emptyset$
  ProposalSearch($\mathbf{s_0}$)

4 EXPERIMENTS

In this section, the results from two experiments are reported: 1) the performance of our reinforcement learning based temporal proposal algorithm, 2) how the salient proposals affect the performance on video action localization.

4.1 Configurations

We use labeled untrimmed videos from Thumos 2014 [9] as the training and testing data for both experiments. Thumos contains totally 2584 untrimmed long videos with temporal annotations of 24 actions. In the experiments for both training and testing, videos without the temporal annotations are eliminated. For the salient proposal task, we label all the temporal regions with action annotations as salient proposals. To increase the convergence speed for the reinforcement learning, we pretrain the sub-Q-network in a traditional fashion as a binary classification model where the regions with larger IoU scores with any of the ground-truth proposals are the positive samples, otherwise are the negative samples.

We evaluate our method in terms of both precision and recall. If a predicted temporal region has a IoU score higher than a certain threshold with a ground-truth region, we consider the proposal a true proposal and the ground-truth region is found. In our experiments, IoU threshold of 0.5 and 0.7 are used. Note that precision and recall are contradictory in some sense. A higher precision tends to result in a lower recall and vice versa. Therefore, we additionally choose F1 score to evaluate the overall performance on both perspectives which is the harmonic mean of the precision and recall: $F1 = 2 * \frac{precision * recall}{precision + recall}$. For action localization, the macro average of the precision, recall and F1 for each action category is computed.

4.2 Salient Proposal Detection

In this section, we report the experiment results on salient proposal detection. In order to have a fair comparison, and verify the effectiveness of the reinforcement learning compared to the other proposal methods, we rule out other factors that could potentially affect model's performance. As a result, all the comparison methods use the same training set and same network structure for feature extraction and proposal detection introduced in section 3. The comparison methods in our experiments are listed as follows:

Pyramid Window: This method is essentially our model without the localizer component. At each search step, rather than using the localizer to select the sub-regions to conduct the further search, this method continues the search in all the sub-regions. As a result, all the temporal regions in the hierarchical structure are fed into the proposal detector. We choose this comparison method to observe how our method performs without the localizer component.

Sliding Window [15]: A set of sliding windows with the time length of 256, 128, 64, 32 and 16 frames are used on each video. All the video clips within the windows are fed into the proposal detector. We choose this method to compare the proposed method with the state-of-art method in video proposal detection.

Table 1 reports the proposal detection performance of our method and the comparison methods. The parameters are tuned with a validation set and the highest F1 score on validation set is selected. The IoU threshold is set to 0.5 and 0.7, and the results of both settings are reported respectively. Overall, compared to the baseline methods, when setting 0.5 and 0.7 IoU threshold, our method achieves the highest F1 score, which verifies our method's overall performance and shows our method's ability to accurately locate a salient proposal without generating too many false positives.

Compared to the pyramid method, our method has a higher precision and overall F1 score on both 0.5 and 0.7 IoU thresholds, which validates that the localizer is a crucial component for our model. The precision improvement from the pyramid method verifies the localizer learned by reinforcement learning is able to help detector to localize the salient proposal more accurately, hence a higher precision. The reason that the pyramid method achieves better recall is that our method uses localizer on top of the pyramid windows, which has the ability to filter out regions without salient proposals, but in the meantime rules out few true positive regions.

Recall is an important criterion in salient proposal detection. For the detected salient proposals, trading a relatively higher false positive rate for finding more ground-truth proposals is acceptable, because the precision of the action localization results can be improved by the future more precise action classification. However,

Table 1: The performance comparison of salient proposal detection in terms of precision, recall and F_1 scores on Thumos 2014 untrimmed videos. The results with the IoU threshold 0.5 and 0.7 are reported respectively. The results shown in boldface are the best results.

IoU Threshold	0.5			0.7		
	precision	recall	F1	precision	recall	F1
Pyramid	0.1426	**0.5887**	0.2296	0.0440	**0.4672**	0.0804
Sliding Window[15]	0.2625	0.5011	0.3445	0.0832	0.2840	0.1287
Our Method	**0.2713**	0.5674	**0.3671**	**0.0873**	0.4401	**0.1457**

high precision in salient proposal detection also helps to rule out more false positives and increases precision for the action localization. Our method achieves comparable recall on both IoU thresholds while maintaining a better precision and F1 score. Although the recall is slightly lower than the pyramid method, our method gains a great precision improvement. Compared to the state-of-art sliding window method, our method manages to get a higher recall with a higher precision.

By using the localizer to search for the salient proposals, the average number of video regions fed into the proposal detector for a single video is effectively decreased. Figure 4 reports the number of average regions needed to be fed into the detector, when the methods achieve the performance in Table 1. As you can see in Figure 4, in our method the detector scans much less video regions compared to sliding window and pyramid method. From Table 1 and Figure 4, we observes that even our method generates much fewer candidates windows for the proposal detector, it achieves a comparable recall, which shows that the localizer is able to reduce the candidate windows by precisely ruling out the temporal regions not containing salient proposals.

To illustrate how the localizer localizes the salient proposals step by step, we select an example from our experiment results, shown in Figure 5. Two salient proposals are localized in this video remarked as blue and orange bounding boxes respectively. In the first iteration, the localizer selects the middle (blue) and right (orange) boxes to continue the further search. In the second iteration, the video region in the blue box showing a soccer player performing a penalty kick is emitted as the salient proposal without further search iteration, and the right most orange bounding box is selected for the further search iteration. In the third iteration, the video region in the orange box showing a goal keeper trying to block a penalty kick is emitted as the salient proposal.

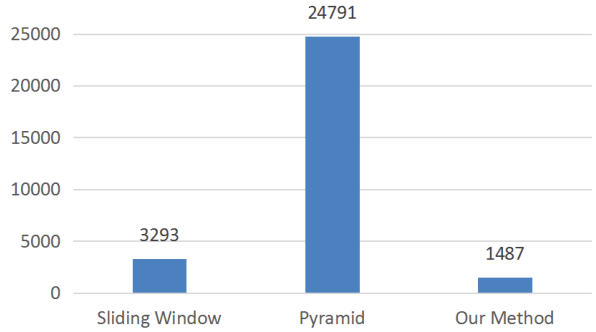

Figure 4: Number of regions fed in to the action detectors, when the performance in Table 1 is achieved.

4.3 Action Localization

In this section, we utilize the salient proposals in one of the most common applications in video analysis, i.e. the action localization. State-of-art approaches train action classifier to recognize actions based on the salient proposals. In our experiments, we train a C3D network for action classification, and feed salient proposals generated by our method and other comparison salient proposal method to verify their effectiveness in action localization.

The comparison salient proposal methods in our experiments are listed as follows:

Pyramid: We directly feed the video regions generated by a pyramid structure into the action classifier. This comparison method will show how well our model performs without the localizer and proposal detector.

Pyramid + Localizer: We directly feed the video regions generated by the localizer into the action classifier. This method will show how well our model performs without the proposal detector.

Pyramid + Proposal Detector: The proposal detector is used on the pyramid windows to generate video regions that potentially contain actions for the action detector. This method will show how well our model performs without the localizer.

Sliding Window [7]: This is a state-of-art approaches commonly used in the action localization competitions. Video regions generated by sliding window are fed into the action detector.

Sliding Window + Proposal Detector [15]: This is a state-of-art approach commonly used in the action localization. A proposal detector is used to generate video regions that potentially contain actions for the action detector.

Based on the experiments results reported in Table 2, we draw following conclusions.

Overall, the proposed method achieves the best overall F1 score on both 0.5 and 0.7 IoU thresholds. This observation validates the reinforcement based salient proposal's effectiveness on action localization. The performance improvement from pyramid method + proposal verifies the effectiveness of the localizer component. The performance improvement from pyramid + localizer method verifies the effectiveness of the detector component.

Methods without proposal network usually achieve higher recall. The pyramid window method has the highest recall on 0.5 IoU threshold and the sliding window method has the highest recall on 0.7 IoU threshold. This is due to the fact that without the

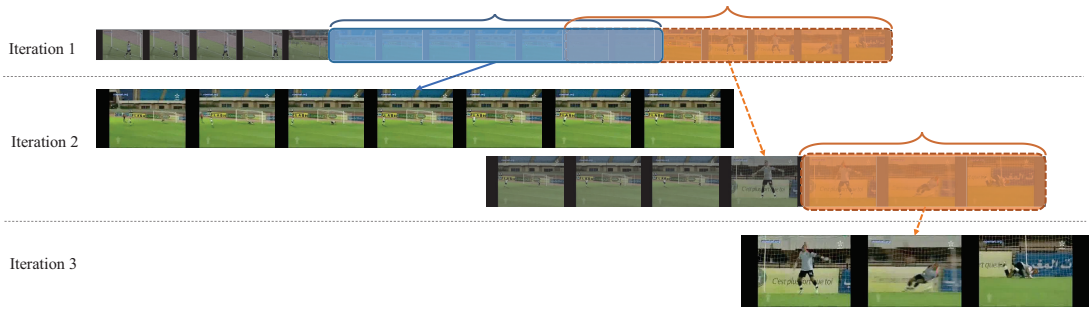

Figure 5: Demonstration of the localizer's search process. Two salient proposals are localized in this video remarked as blue and orange boxes respectively.

Table 2: The performance comparison of action localization in terms of precision, recall and F1 scores on Thumos 2014 untrimmed videos. The results with the IoU threshold 0.5 and 0.7 are reported respectively. The results shown in boldface are the best results.

IoU Threshold	0.5			0.7		
	precision	recall	F1	precision	recall	F1
Pyramid	0.0686	**0.5305**	0.1214	0.0276	0.2629	0.0499
Sliding Window [7]	0.0638	0.5168	0.1136	0.0179	**0.3670**	0.0342
Pyramid + Proposal Detector	0.1226	0.3494	0.1815	0.0322	0.2533	0.0571
Sliding Window + Proposal Detector[15]	0.1760	0.2753	0.2147	0.0533	0.1497	0.0786
Pyramid + Localizer	0.1475	0.4587	0.2233	0.0443	0.3602	0.0789
Our Method	**0.1909**	0.3408	**0.2447**	**0.0549**	0.2410	**0.0894**

proposal network filtering out the background regions, it generates a lot more proposals for the action classification, so it is more likely for methods without proposal network to emit more true positives. However, the proposal network has the ability to filter out the backgrounds temporal regions, so the methods applying proposal detector achieve higher precision and F1 score than the ones without proposal network. For example, Pyramid and sliding window both have obvious precision and F1 score improvement when applying the proposal detector. On the other hand, although without proposal detector, the pyramid + localizer method achieves high recall while maintaining a relatively comparable recall rate of 0.1475, which shows that besides accurately localizing salient proposals, the localizer also has the ability to filter out the background temporal regions without losing too many true positive regions in the process.

5 CONCLUSIONS

This paper proposes a reinforcement learning based video temporal proposal method that learns a search policy that instead of exploring every video segment, finds an optimal search path to locate a salient proposal based the currently observing video segments. The model consists of two collaborative components, namely the localizer and the detector. The localizer searches for the temporal regions potentially containing salient proposal in a hierarchical structure. It follows a search policy learned by reinforcement learning and iteratively selects the next sub-region to continue the search at each step. The detector is responsible for emitting the final salient proposal results. From the experiments, we can observe that compared to the state-of-art method, by adapting reinforcement learning, our method is able to localize salient proposals with fewer candidates windows and more precise results with a comparable recall.

6 ACKNOWLEDGEMENTS

This work was supported in part by NSFC (No.U1509206, U1611461, 61625107), 973 program (No. 2015CB352302), Chinese Knowledge Center of Engineering Science and Technology (CKCEST), Qianjiang Talents Program of Zhejiang Province 2015, Key program of Zhejiang Province (2015C01027).

REFERENCES

[1] Miriam Bellver, Xavier Giro-i Nieto, Ferran Marques, and Jordi Torres. 2016. Hierarchical Object Detection with Deep Reinforcement Learning. In *Deep Reinforcement Learning Workshop, NIPS*.
[2] Juan C Caicedo and Svetlana Lazebnik. 2015. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 2488–2496.
[3] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. 2014. BING: Binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3286–3293.
[4] jifeng dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *Advances in Neural Information Processing Systems 29*. 379–387.
[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

[6] Georgia Gkioxari and Jitendra Malik. 2015. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 759–768.

[7] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–970.

[8] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees GM Snoek. 2014. Action localization with tubelets from motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 740–747.

[9] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. 2014. THUMOS Challenge: Action Recognition with a Large Number of Classes. http://crcv.ucf.edu/THUMOS14/. (2014).

[10] Zequn Jie, Xiaodan Liang, Jiashi Feng, Xiaojie Jin, Wen Lu, and Shuicheng Yan. 2016. Tree-Structured Reinforcement Learning for Sequential Object Localization. In *Advances in Neural Information Processing Systems*. 127–135.

[11] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu. 2016. Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2894–2902.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[13] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. 2013. Action and event recognition with fisher vectors on a compact feature set. In *Proceedings of the IEEE international conference on computer vision*. 1817–1824.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28*. 91–99.

[15] Zheng Shou, Dongang Wang, and Shih-Fu Chang. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1049–1058.

[16] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems 27*. 568–576.

[17] Chen Sun, Sanketh Shetty, Rahul Sukthankar, and Ram Nevatia. 2015. Temporal Localization of Fine-Grained Actions in Videos by Domain Transfer from Web Images. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. 371–380.

[18] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[19] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. 2013. Selective Search for Object Recognition. *International Journal of Computer Vision* 104, 2 (2013), 154–171.

[20] Heng Wang and Cordelia Schmid. 2013. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*. 3551–3558.

[21] Zhongwen Xu, Yi Yang, and Alex G Hauptmann. 2015. A discriminative CNN video representation for event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1798–1807.

[22] Jimei Yang and Ming-Hsuan Yang. 2012. Top-down visual saliency via joint CRF and dictionary learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. 2296–2303.

[23] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. 2015. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738* (2015).

[24] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. 2016. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2678–2687.