applied sciences

MDPI

Article

Temporal Action Detection in Untrimmed Videos from Fine to Coarse Granularity

Guangle Yao [1,2,3], Tao Lei [1,*], Xianyuan Liu [1,3] and Ping Jiang [1]

[1] Institute of Optics and Electronics, Chinese Academy of Sciences, P.O. Box 350, Shuangliu, Chengdu 610209, China; guangle.yao@std.uestc.edu.cn (G.Y.); liuxianyuan16@mails.ucas.ac.cn (X.L.); jiangping@ioe.ac.cn (P.J.)

[2] School of Optoelectronic Information, University of Electronic Science and Technology of China, No. 4, Section 2, North Jianshe Road, Chengdu 610054, China

[3] University of Chinese Academy of Sciences, 19 A Yuquan Rd, Shijingshan District, Beijing 100039, China

* Correspondence: taoleiyan@ioe.ac.cn; Tel.: +86-177-1685-3605

Received: 10 August 2018; Accepted: 9 October 2018; Published: 15 October 2018

check for updates

Abstract: Temporal action detection in long, untrimmed videos is an important yet challenging task that requires not only recognizing the categories of actions in videos, but also localizing the start and end times of each action. Recent years, artificial neural networks, such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) improve the performance significantly in various computer vision tasks, including action detection. In this paper, we make the most of different granular classifiers and propose to detect action from fine to coarse granularity, which is also in line with the people's detection habits. Our action detection method is built in the 'proposal then classification' framework. We employ several neural network architectures as deep information extractor and segment-level (fine granular) and window-level (coarse granular) classifiers. Each of the proposal and classification steps is executed from the segment to window level. The experimental results show that our method not only achieves detection performance that is comparable to that of state-of-the-art methods, but also has a relatively balanced performance for different action categories.

Keywords: action detection; action proposal; convolutional neural network; regression network

1. Introduction

Video analysis is important for applications ranging from robotics, human-computer interaction to intelligent surveillance. It has attracted extensive research attention in computer vision and artificial intelligence communities. Action recognition and action detection are two important branches of video analysis. The former aims to classify the categories of actions in manually trimmed videos. The latter is more challenging, which requires not only recognizing the categories of actions, but also localizing the start and end times of each action in long, untrimmed videos. In this paper, we address the action detection problem.

Recently, the progression of action detection parallels the development of object detection and it has undergone significant advancements. Most of the action detection methods are fell into four types of frameworks, as illustrated in Figure 1. The first framework [1,2] performs frame-level or segment-level classification, and then applies post-processing to merge the predications into detection results. The second framework 'proposal then classification' [3,4] draws inspiration from the Region-based Convolutional Neural Networks (R-CNN) object detection [5] and its upgraded versions [6,7]. It is implemented in two steps: (1) temporal action proposals, which produces a set of windows that are likely to contain an action instance; and, (2) action classification, which provides the specific category of the action proposal. The third framework [8,9] is a unified end-to-end 'single-stream' action detection framework, which is motived by single-shot object detectors such as YOLO [10,11] and SSD [12]. The last framework [13,14] is enlightened by the deconvolution layers in the fully-convolutional network-based object segment [15] and object detection [16,17]. It employs deconvolution for temporal upsampling to ensure that the temporal length of output be same with the input video and predicts actions at the frame-level.

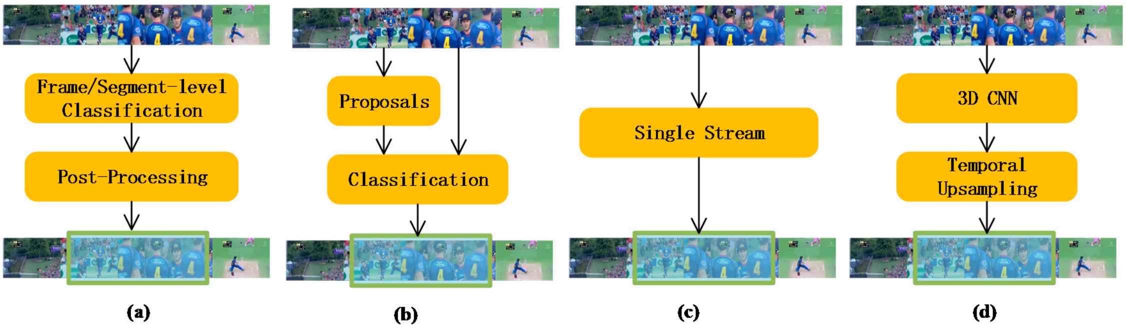


Figure 1. There have been four dominant frameworks for temporal action detection: (**a**) classification then post-processing; (**b**) proposal then classification; (**c**) single stream; and, (**d**) temporal upsampling.

Most of these mentioned action detection methods design fine granular (frame-level and segment-level) or coarse granular (window-level and anchor-level) classifiers to perform action proposal, classification or localization. For examples, Shou et al. [3] trained three three-dimensional (3D) CNN classifiers, respectively, for segment-level proposal, classification, and localization; Zhu et al. [4] trained a 3D CNN classifier via Multi-task learning method for segment-level proposal, classification, and localization. Both of the methods [3,4] used post-processing to obtain the final detection results. In contrast, Buch et al. [9] and Gao et al. [18] used off-the-shelf (pre-trained) 3D CNN to extract segment-level feature without classification, and performed the window-level classification as the detection results. The trade-off between the fine and coarse granular classifiers is that the fine granular classifier tends to offer precise temporal boundaries of actions, while the coarse granular classifier considers the dependence between the frames or segments of one action instance. This motivates us to design both of the fine and coarse granular classifiers to improve the performance of action detection. As illustrated in Figure 2, we build our method in the 'proposal then classification' framework. Each of the proposal and classification steps is executed from fine (segment-level) to coarse (window-level) granularity. We split the videos into non-overlapped segments. In the proposal step, we first train Res3D [19], a powerful 3D CNN architecture, as a binary classifier for the segments of videos. Then, we propose a discriminative temporal selective search to generate candidates of window-level proposal, and design a regression network via multi-task learning to classify the proposal candidates into proposals and refine their tempor bounraries. In the classification step, we first train Res3D as a multi-class classifier for the segments of proposals. Then, we employ Long Short-Term Memory (LSTM) [20] on the features of Res3D to classify the proposal into specific action category. In summary, this paper makes the following contributions:

1. We propose to detect action from fine to coarse granularity, and build our action detection method in the 'proposal then classification' framework.
2. The proposed and designed components in our method, such as: a Res3D version for the segment with 16 frames, transfer learning from RGB to optical flow, discriminative selective search, regression network via multi-task learning can be used in other action recognition or detection methods.
3. The experimental results show that our method achieves state-of-the-art performance on THUMOS'14 and has a relatively balanced performance for different action categories.

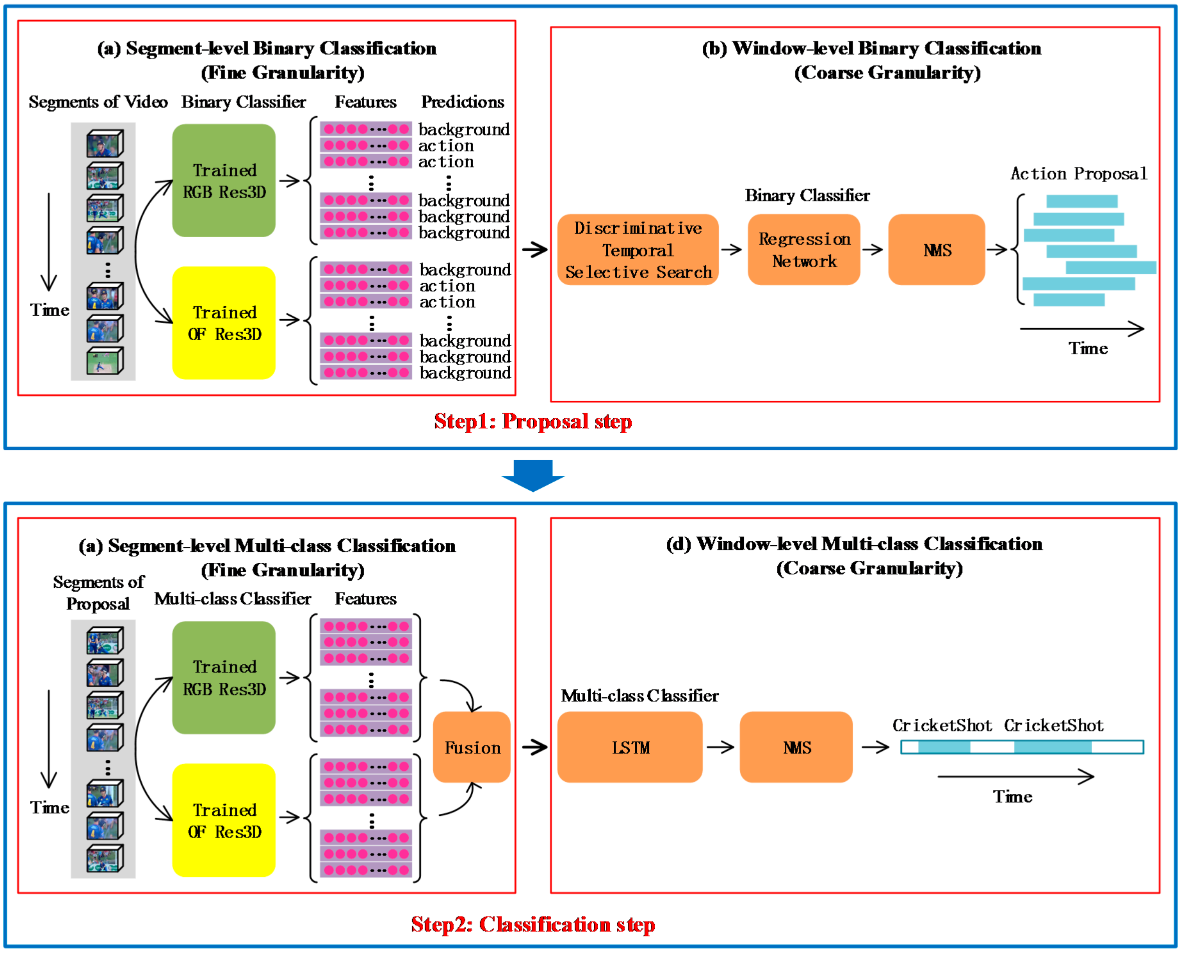


Figure 2. Overview of our proposed detection method. In each of proposal and classification steps, our method performs classification from segment-level to window-level.

The remainder of this paper is organized as follows: Section 2 describes previous action detection studies. Section 3 presents our proposed action detection method. Section 4 provides the details of experiments and discusses the experimental results. Conclusions and future work are given in Section 5.

2. Previous Works

Temporal action detection in videos is an active area of research. The earlier works focus on hand-crafted feature representations for action. Recently, much progress of action detection has been facilitated by convolutional neural network (CNN) feature. Meanwhile, the methodologies of spatial object detection were extended to temporal action detection. As described in Section 1, most of the prior action detection methods fall into four types of frameworks: classification then post-processing, proposal then classification, single-stream, and temporal upsampling. In this section, we outline the major contributions of these frameworks.

Classification then Post-processing. Montes et al. [2] used the C3D [21], a type of 3D Convolutional Neural Network architecture, to extract features of video segments and trained LSTM to classify the segments. Finally, they determined the temporal boundaries of actions by post-processing. Wang et al. [22] firstly conducted temporal sliding window scanning to split a video into short segments. Then, they adopted CNN features and the Fisher Vector [23] of improved Dense Trajectories (iDT) [24] to represent each segment. Finally, they trained multiple one-vs-all Support Vector Machine (SVM) to perform segment-level recognition and generated the detection results using the predictions of segments.

Proposal then Classification. Shou et al. [3] exploited three segment-level C3D networks to address the problems of proposal, classification and localization, respectively, and used the classification network to serve as initialization for the localization network. Zhu et al. [4] trained a C3D via multi-task learning to perform proposal, classification and localization in parallel. Xu et al. [25] computed fully-convolutional 3D CNN features and proposed temporal regions, then conducted pooling of the features within these 3D regions to predict the categories of actions. Also some works [18,26,27] focused on generating high quality action proposals. Escorcia et al. [26] and Buch et al. [27] used LSTM networks to encode a video stream and produced proposals inside the video stream. Gao et al. [18] proposed to jointly predict action proposal and to refine the temporal action boundary by temporal coordinate regression.

Single-stream. Buch et al. [9] trained deep recurrent architecture based on enforcing semantic constraints on intermediate modules that are gradually relaxed as learning progresses, and processed the video in a single stream, which directly outputs the temporal boundaries of actions and corresponding action categories. Lin et al. [8] extracted snippet-level action score features via CNN and 3D CNN networks and designed a single shot action detector network, in which base layers are used to reduce the temporal dimension of the input data; anchor layers output multiple scale feature map associated with anchor instances and prediction layers are used to predict the categories, locations, and confidences of anchor instances.

Temporal Upsampling. Shou et al. [13] designed a Convolutional-De-Convolutional (CDC) network that places CDC filters on top of C3D network. The proposed CDC filter performs temporal upsampling and spatial downsampling simultaneously to predict actions at the frame-level. Yang et al. [14] proposed a Temporal Preservation Convolution (TPC) network that equips C3D with TPC filters. TPC filter can fully preserve the temporal resolution and downsample the spatial resolution simultaneously, enabling frame-level action localization with the minimal loss of time information.

3. Our Method

Our action method is built in the 'proposal then classification' framework. For the purpose of detection from fine to coarse granularity, we design the segment-level and window-level classifiers for each of the proposal and classification steps. As illustrated by Figure 2, in the proposal step, we train Res3D and regression network, respectively, as the binary segment-level and window-level classifiers. In the classification step, we train Res3D and LSTM, respectively, as the multi-class segment-level and window-level classifiers.

In this section, we first present an overview of our method, and then describe its key components, including Res3D architecture, discriminative temporal selective search, and regression network.

3.1. Overview of Our Method

Given a video $V = \{v_i\}_1^L$ with L frames, we split it into num consecutive non-overlapped segments, where $num = L/n$, and n is the frame number of a segment. We also denote the video V as a segment set: $S = \{s_i\}_1^{num}$. In our method, we use the segments with $n = 16$ frames.

In the proposal step, we modify the Res3D for 16 frame-length segments, and train it as a binary classifier to classify the segments of the video into background or action and extract pool5 features of these segments. To impose Res3D on optical flow (OF), we provide some practices to transfer the knowledge from RGB to OF. Then, we propose a discriminative temporal selective search to generate the action proposal candidates with variable lengths. We train a regression network to classify the proposal candidates into proposal and refine their temporal boundaries simultaneously. Finally, we use Non-Maximum Suppression (NMS) with an overlapped threshold 0.3 to remove the redundant proposals.

In the classification step, we train the modified Res3D as a $K + 1$ classes classifier, where K is the number of the action categories, and 1 indicates the category of background. We extract pool5 features from the segments of proposals, and feed them into LSTM network. Then, we use a fully-connected layer on top of the LSTM'S hidden state, followed by a softmax layer, to obtain a score corresponding to each proposal. Finally, we refine this score by multiplying the score of regression network in the proposal step and perform NMS with an overlapped threshold 0.1 to obtain the final detection results.

The implementation of the Res3D architecture, knowledge transfer from RGB to OF, discriminative temporal selective search, and regression network will be described in the rest of this section. The training details of the Res3D architectures, regression networks, and LSTM will be described in Section 4.2.

3.2. Res3D Architecture

Convolutional neural network (CNN) has achieved remarkable performance in tasks of image and video domains. In our method, we employ Res3D [19], a ResNet18-style [28] 3D CNN architecture, as the segment-level classifier and feature extractor.

In the past several years, various two-dimensional (2D) and 3D CNN architectures have been designed by the computer vision community, including AlexNet [29], ZFNet [30], VGGNet [31], GoogLeNet [32], ResNet [28], C3D [21], and Res3D [19], etc. The models (weights) for these CNN architectures were obtained by pre-training on large-scale datasets, typically the ImageNet dataset [33] for 2D CNN architectures and the Sports-1M dataset [34] for 3D CNN architectures. For a new small-scale dataset or a new modality, transfer learning is performed to fine-tune the pre-trained model.

Res3D, which employs the residual connections [28] in 3D CNN, has achieved the best performance among the available 3D CNN architecture to extract the motion information of actions. It is primarily designed for video segments with eight frames. Videos are split into non-overlapped segments with 8 frames as the input of Res3D, and a softmax layer outputs the recognition results for each segment. The original Res3D work [19] provides a pre-trained model that was trained on the Sports-1M dataset. Here, we investigate Res3D architectures for segments with 16 frames using the pre-trained model and transfer the knowledge learned from RGB to optical flow.

For simplicity, we omit the batch size, and denote the shape of the input for each layer as 4D tensors $C \times T \times H \times W$, where C, T, H, and W are the channel, temporal length, height, and width, respectively. We observe that the full-connected (FC) layer in Res3D still works in 2D space, and the temporal depth T of the full-connect layer's input should be 1. Thus, Res3D cannot be used to extract information directly from the segment of 16 frames. To address this isuue, Tran et al. [19] conducted a frame temporal sampling on the segment with 16 frames in the input layer of Res3D. However, this method drops the information in the even frames. Instead of the temporal sampling in the input layer, we propose to add temporal down-sampling into the building blocks conv2_x by setting the temporal_stride of its branch1 and branch2a to 2. We name the Res3D used directly on segment with 16 frames Res3D-16D; name the Res3D with temporal frame sampling in the input layer Res3D-16S. The shapes of each building block's inputs for these architectures are presented in Table 1.

Table 1. Res3D architectures of different Res3D versions.

Layers	**Building Blocks**	**Shape of The Input**			
		Res3D	**Res3D-16D**	**Res3D-16S**	**Our Res3D**
Conv1	$3 \times 7 \times 7.64$	$3 \times 8 \times 112 \times 112$	$3 \times 16 \times 112 \times 112$	$3 \times 8 \times 112 \times 112$	$3 \times 16 \times 112 \times 112$
Conv2_x	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 2$	$64 \times 8 \times 56 \times 56$	$64 \times 16 \times 56 \times 56$	$64 \times 8 \times 56 \times 56$	$64 \times 16 \times 56 \times 56$
Conv3_x	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 2$	$64 \times 8 \times 56 \times 56$	$64 \times 18 \times 56 \times 56$	$64 \times 8 \times 56 \times 56$	$64 \times 8 \times 56 \times 56$
Conv4_x	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 2$	$128 \times 4 \times 28 \times 28$	$128 \times 8 \times 28 \times 28$	$128 \times 4 \times 28 \times 28$	$128 \times 4 \times 28 \times 28$
Conv5_x	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 2$	$256 \times 2 \times 14 \times 14$	$256 \times 4 \times 14 \times 14$	$256 \times 2 \times 14 \times 14$	$256 \times 2 \times 14 \times 14$
FC	InnerProduct, Softmax	$512 \times 1 \times 7 \times 7$	$512 \times 2 \times 7 \times 7$ (Not functional)	$512 \times 1 \times 7 \times 7$	$512 \times 1 \times 7 \times 7$

The optical flow (OF), which is invariant to appearance, is used as the input of the 2D and 3D CNN architectures for action recognition. Simonyan et al. [35] trained 2D CNN from scratch using stacked optical flow. Ji et al. [36] trained 3D CNN from scratch using optical flow. Fine-tuning a pre-trained model has been shown to be effective for transferring the knowledge learned from a large-scale dataset to a new dataset or a new modality. Wang et al. [37] fine-tuned 2D CNN on optical flow using the knowledge learned from large-scale RGB image dataset. We echo these works and propose to fine-tune 3D CNN on optical flow using the knowledge that was learned from a RGB video dataset.

The pre-trained Res3D model provided by Tran et al. [19] takes 3-channel RGB segments as the input of CNN. To transfer the knowledge learned from the RGB to optical flow, we first design 3-channel optical flow. We extract the optical flow using TVL1 algorithm [38] for each video and discretize it by a linear transformation such that the optical flow values fall into the range [0, 255]. This transformation yields 2 channels of optical flow: the horizontal optical flow OF_x and the vertical optical flow OF_y. Then, we calculate the mean of OF_x and OF_y as the third channel. Figure 3 presents two consecutive frames, OF_x, OF_y, and the three-channel optical flow. This step ensures that the optical flow has same number of channels as the RGB used for the input of the Res3D architecture. To transfer the RGB knowledge to optical flow, which is a new modality, we empirically increase learning_rate, step_size and max_iteration during fine-tuning. Compared with RGB Res3D, we increase these variables, by 10 times, five times, and five times in OF Res3D, respectively, to accelerate the model updating. The detailed parameters are described in Section 4.2.

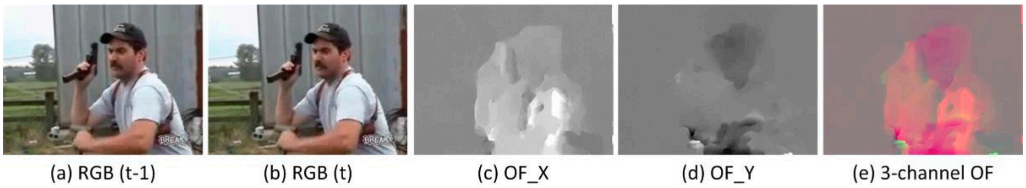


Figure 3. Two consecutive frames, optical flow in the horizontal and vertical directions and 3-channel optical flow.

3.3. Discriminative Temporal Selective Search

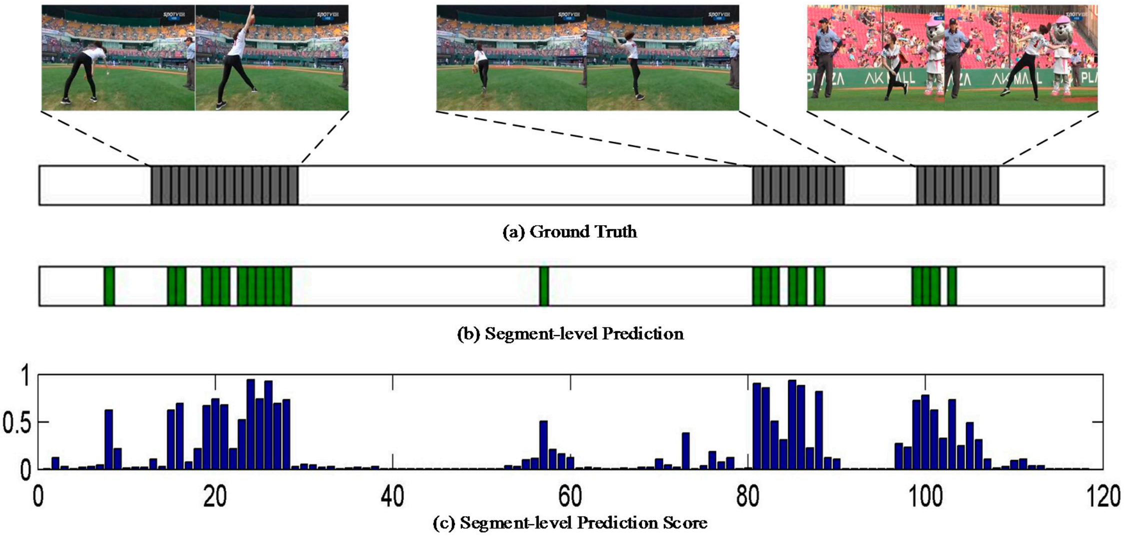
Inspired by the selective search [39] used in RCNN object detection [5], we introduce a temporal selective search to generate window-level action proposal candidates: two adjacent segments with minimum distance are merged as a temporal region, iteratively, until there is only one temporal region, and the regions existed in the merging procedure are output as the action proposal candidates. Furthermore, different with the original selective search that is performed in the non-discriminative feature space, such as color space and text space, our temporal selective search is performed in the discriminative feature space. Each segment is predicted into background or action and its feature is extracted by binary Res3D classifier. An example of the segment-level prediction is illustrated in Figure 4. We denote a segment as an action segment if it is predicted into action by RGB or OF Res3D. We can take advantage of this discriminability of the segments in temporal selective search to improve the quality of the proposal candidates by designing three new principles in the procedure of temporal selective search:

1. Merge Principle: In each iteration, the merged segment or temporal region should contain action segment.
2. Keep Principle: For a temporal region, if the percentage of predicted action segments is below a threshold θ, it would be excluded from the set of proposal candidates.
3. Stop Principle: The merging is stopped until there is only one temporal region containing action segments.

We name this proposed method Discriminative Temporal Selective Search, and detail it in Algorithm 1. For a complementary similarity measurement of segments, we use L2 distance on the deep feature of the GRB and OF Res3D architectures. The similarity measurement of two segments s_i and s_j is a combination of two measurements:

$$sm(s_i, s_j) = sm_{RGB}(s_i, s_j) + sm_{OF}(s_i, s_j), \tag{1}$$

The threshold θ in the Keep Principle is set to 0.4. We select the temporal regions with length between 2 and 50 segments as the proposal candidates, which are feed into a regression network for the further processing.



(a) Ground Truth

(b) Segment-level Prediction

(c) Segment-level Prediction Score

Figure 4. The segment-level prediction by the binary RGB Res3D classifier. The testing video video_test_0000964 is split into 118 segments, and each segment is predicted into action or background.

Algorithm 1. Discriminative temporal selective search

```
Input: the video $V$ and its segments set $S = \{s_i\}_1^{num}$
          discriminative segment-level features set $F = \{f_i\}_1^{num}$
          segment-level prediction set $PR = \{pr_i\}_1^{num}$
Output: Set of the temporal action proposal candidates $PC$
Initialize similarity set $SM = \varnothing$
Initialize proposal candidates set $PC = \varnothing$
foreach consecutive segment pair $(s_i, s_j)$ do
    Calculate similarity $sm(s_i, s_j)$;
    $SM = SM \cup sm(s_i, s_j)$;
while more than one $s_i$ contains action segment do
    Get the highest similarity $sm(s_i, s_j) = \max(SM)$, where $s_i$ or $s_j$ contains action segment;
    Merge $s_i$ and $s_j$: $s_{merge} = s_i \cup s_j$;
    Merge the features of $s_i$ and $s_j$: $f_{merge} = mean(f_i, f_j)$;
    Calculate similarity set $SM_{merge}$ between $s_{merge}$ and its consecutive segments or regions;
    Update similarity set: $SM = (SM \backslash (sm(s_i, s_*) \cup sm(s_*, s_j))) \cup SM_{merge}$;
    Update features set: $F = (SM \backslash (f_i \cup f_j)) \cup f_{merge}$;
    Update segments set: $S = (S \backslash (s_i \cup s_j)) \cup s_{merge}$;
    if the percentage of action segments in $s_{merge}$ beyond threshold $\theta$ then
    $PC = PC \cup s_{merge}$;
```

3.4. Regression Network

Motivated by Fast R-CNN [6] where a regression network is incorporated to refine the bounding box of the predicted object, we introduce a regression network for the temporal boundary refinement of the proposals, which is illustrated in Figure 5. Further, we train the regression network via multi-task leaning. It is also used as a window-level classifier to predicate the proposal candidates into proposals.

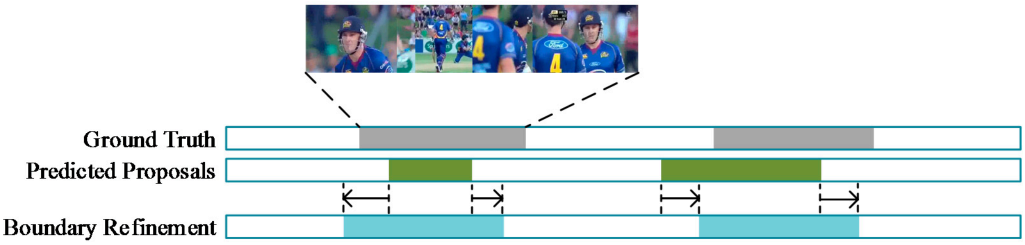


Figure 5. The illustration of boundary refinement.

Given a proposal candidate p, which is composed of segments $\{s_i\}_s^e$ with features $f_{RGB} = \{fr_i\}_s^e$ and $f_{OF} = \{fo_i\}_s^e$, where s and e are the indices of staring and end segments of the proposal candidate p, fr_i and fo_i are the pool5 features of segment s_i, respectively, as extracted by the trained RGB and OF Res3D. The representation of the proposal candidate p is given by:

$$Concatenation(AveL2(F_{RGB}), AveL2(F_{OF})), \tag{2}$$

where $AveL2(*)$ represents the operation of averaging and L2-normalization. The regression network takes the representation of proposal candidate as input, and outputs the confidence score, indicating whether the input proposal candidate is an action proposal, and also outputs the temporal boundary regression offsets, which are defined as follows:

$$o_s = s_p - s_g, \tag{3}$$

$$o_e = e_p - e_g, \tag{4}$$

where s_p and e_p are the starting and ending segment indices of the proposal; s_g and e_g are the starting and ending segment indices of the matched ground truth. We employ multi-task loss L to jointly train the classifier and temporal boundary regression.

$$L = L_{cls} + \lambda L_{reg}, \tag{5}$$

where L_{cls} is a standard binary softmax cross-entropy loss for action/background classification. L_{reg} is the temporal regression loss, which is defined as:

$$L_{reg} = \frac{1}{N_p} \sum_{i=1}^{N_p} l_i(|o_{s,i}|+|o_{e,i}|), \tag{6}$$

where l_i is the label, 1 for positive (action) sample and 0 for negative (background) sample. N_p is the number of positive samples. In other words, we only regress the boundary of positive samples. We utilize L1 norm to make the loss be more robust to outliers. The strategy to construct training samples is described in Section 4.2.

Additionally, the predication score of the proposal is also used to evaluate the proposal and to refine the action categorization score in the classification step.

4. Experiments

4.1. Datasets and Evaluation Metrics

Both of the datasets THUMOS'14 [40] and ActivityNet1.3 [41] can be used to evaluate multiple action-related tasks, including action detection. THUMOS'14 is the most widely used benchmark in the action detection works, and to the best of our knowledge, the ActivityNet1.3 is the largest existing dataset for action detection task. THUMOS'14 includes the training set, validation set, background set, and testing set. Training set and validation set consist of trimmed and untrimmed videos, respectively, with 20 action categories. Background set contains videos that do not include any action. Testing set includes untrimmed videos. Some of the videos might contain one or multiple action instances from one or multiple action categories. ActivityNet1.3 contains 19,994 untrimmed videos in training set, validation set and testing set, with 200 action categories.

In the experiments, we perform the evaluation of binary and multi-class Res3D classifiers with the metric: segment-level recognition accuracy. For the action proposal evaluation, we follow the standard metrics in the temporal action proposal task of ActivityNet Challenge [41]: Average Recall vs. Average Number of Proposal per Video (AR-AN) Curve and Metric Score, which is the area under the resulting final curve. We port the official evaluation toolkit of ActivityNet to THUMOS'14. We also use the metric: Recall @ 1K proposal vs. tIoU (Recall@1K-tIoU) plot [26] to evaluate the action proposal. The action detection is evaluated using Mean Average Precision (mAP) at 0.5 tIoU. We use the official evaluation toolkit provided by THUMOS'14 challenge.

4.2. Implementation Details

Res3D training: Res3D architectures are trained on the training, validation, and background sets. We generate non-overlapped segments by uniformly sampling 16 frames. In the proposal step, for each segment of the trimmed videos in training set, we set its label as positive. For the untrimmed videos in validation set, we assign the segments of ground truth with positive labels and assign the segments which do not overlap with a ground truth with negative labels. We also assign each segment of background videos with negative label. At last, we obtain $N_{trn} + N_{vld}$ positive samples and also randomly selected $N_{trn} + N_{vld}$ negative samples for training. In the classification steps, we follow the similar segments generation strategy. Except when assigning label for positive segment, we explicitly indicate specific action category. In order to balance the number of training data for each category, we randomly select $\frac{N_{trn}+N_{vld}}{K}$ negative samples for training. In the proposal step, for RGB Res3D, learning rate begins at 0.001, decreases by $1/10$ every epoch, and stops at four epochs; for OF Res3D architecture, the learning rate starts at 0.01, decreases by $1/10$ every five epochs, and stops at 20 epochs. In the classification steps, for RGB Res3D, learning rate begins at 0.0001, decreases by $1/10$ every epoch, and stops at four epochs; for OF Res3D architecture, the learning rate starts at 0.001, decreases by $1/10$ every five epochs, and stops at 20 epochs.

Regression network training: Regression network is trained on the validation set, as the videos in the validation set of THUMOS'14 is untrimmed. We use multi-scale sliding windows to generate 50% overlapped windows with variable lengths of $\{2, 3, \ldots, 50\}$ segments as the training samples. We assign a positive label to a window if: (1) the window with the highest temporal Intersection over Union (tIoU) overlaps with a ground truth; or, (2) the window has tIoU larger than 0.5 with any of the ground truth. We assign a negative label to a window which has no overlap with any ground truth. During training, the learning rate and batch size are set as 0.005 and 128, respectively. The ratio of sample numbers of background to action in a batch is set to 10. λ is set to 2.

LSTM training: LSTM is trained on the training, validation and background sets. We use each trimmed video in the training set and each action instance in the validation set for training. We set their labels as the specific action category and obtain $M_{trn} + M_{vld}$ action samples. For background samples, we generate windows with 2 to 50 segments from the validation and background sets. The windows from validation set are not overlapped with any ground truth. We randomly select

$\frac{M_{trn}+M_{vld}}{K}$ background samples. During training, we set the learning rate to 0.0001, and stop the training at 90 epochs.

For the ActivityNet1.3 dataset, we setup the experiments almost following the aforementioned implementation of THUMOS'14. The differences are that there is no background set and the videos in the training set is untrimmed. Thus, we generate the segment-level and window-level training samples from the untrimmed videos in the training and validation sets. In the training and testing phase, we also update the output number of fully-connected layer in the multi-class Res3D and LSTM of the classification step.

4.3. Exploratory Study

We conduct extensive experiments to evaluate the performance of components in our proposed method on THUMOS'14. Firstly, we evaluate the capability of Res3D classifier for segments with 16 frames. The segment-level recognition accuracy in the Table 2 shows that our proposed Res3D outperforms the Res3D-16S [19] for a segment with 16 frames. For the binary classifier in the proposal step, we improve the segment-level recognition accuracy from 68.1% to 78.3%. For the multi-class classifier in the classification step, the accuracy is improved from 67.3% to 69.8%.

Table 2. Segment-level recognition accuracy (%) of different Res3D versions using RGB input field.

Classifiers	Res3D-16D	Res3D-16S	Our Res3D
Binary Classifier	Not Functional	68.1	78.3
Multi-class Classifier	Not Functional	67.3	69.8

In addition, we evaluate our proposed discriminative temporal selective search (DTSS) on the features of the trained Res3D. Sliding window is widely used in action proposal and detection. We compare DTSS with sliding window (SW) and temporal selective search (TSS) on the features of the pre-trained Res3D. We combine these methods with the aforementioned regression network and NMS to form the complete action proposal step and conduct the evaluation of the SW-based, TSS-based, and DTSS-based action proposals. The AR-AN curves of our DTSS-based action proposal are illustrated in Figure 6. The dashed lines are for the recall performance over various tIoU thresholds, while the solid line is the average recall across all tIoU thresholds. For a clear comparison of these three methods, we present the average recall and metric score of the compared methods in Figure 7a and present Recall@1K-tIoU in Figure 7b. These results show that our DTSS-based method performs the best. The SW-based and TSS-based proposal methods only employ window-level classifier. In contrast, our DTSS-based methods employ both of segment-level and window-level classifiers. Thus, these results also support our 'detection from fine to coarse granularity' idea.

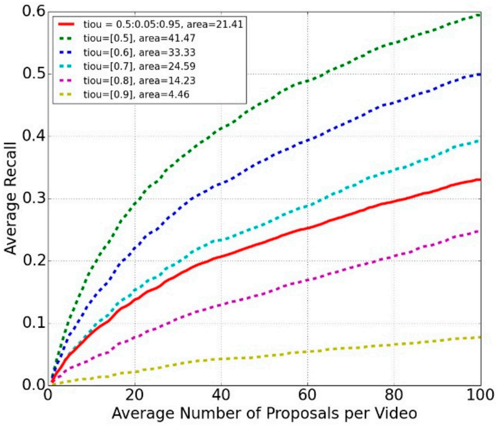
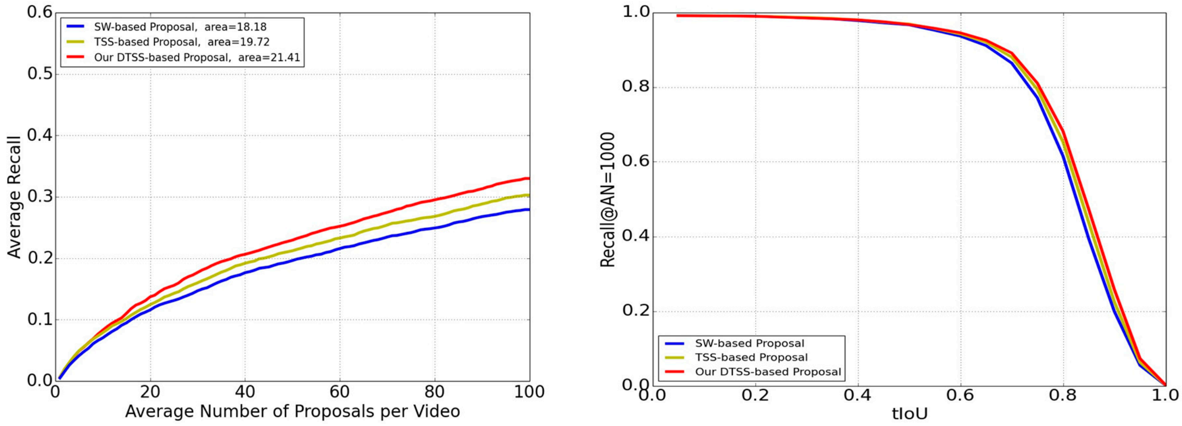


Figure 6. Average Recall vs. Average Number of Proposal (AR-AN) curves of our discriminative temporal selective search (DTSS)-based proposal.



(a) AR-AN Curves **(b) Recall@1K-tIoU Plots**

Figure 7. Comparison of our DTSS-based proposal with the sliding window (SW)-based and temporal selective search (TSS)-based proposals.

In this paper, we provide practices to transfer the knowledge learned by 3D CNN from RGB to optical flow and impose our method on RGB and OF segments. Table 3 presents the results of our action detection using different input fields. The results show that the transfer learning works well and our method benefits from the optical flow field.

Table 3. Action detection results with 0.5 tIoU on THUMOS'14 using different input fields.

Input Fields	mAP (%)
RGB	25.2
OF	24.9
RGB + OF	29.6

4.4. Comparison with State-of-the-Art Methods

We compare the performance of our action detection method with state-of-the-art methods on THUMOS'14, including SCNN, CDC, SS-TAD, and so on. We group these methods according to the aforementioned detection frameworks. In addition to the results at various tIoU thresholds in Table 4, for each method, we also specify the fields of input and the involved features. The results show that our method achieves detection performance that is comparable to that of state-of-the-art methods. Some detection results of two testing videos are presented in Figure 8.

We also compare our method with LearSubmission, SCNN, SS-TAD, and R-C3D for each category in THUMOS'14 at 0.5 tIoU. The Average Precision (AP) for each category is shown in Figure 9. It is noticed that our method performs best for two out of 20 action categories. Both of SS-TAD and R-C3D perform best for seven of 20 action categories. But, our method has the best overall performance among these methods, even though it only outperforms R-C3D and SS-TAD with a small margin. These results indicate that our method has a relatively balanced performance for different action categories.

Table 4. Comparison with state-of-the-art methods on THUMOS'14: mAP (%) with various tIoU thresholds (0.5, 04, 03, 02, 0.1) on THUMOS'14.

Methods	0.5	0.4	0.3	0.2	0.1	Input Fields	Involved Features
Classification then Post-Processing							
UnifiSubmission [42]	0.9	1.4	2.1	3.4	4.6	RGB + OF	iDT
PSDF [1]	18.8	26.1	33.6	42.6	51.4	RGB + OF	iDT
CUHKSubmission [22]	8.3	11.7	14.0	17.0	18.2	RGB + OF	iDT + CNN
Temporal Upsampling							
CDC [13]	23.3	29.4	40.1	-	-	RGB	3D CNN
TPC [14]	28.2	37.1	44.1	-	-	RGB	3D CNN
Single Stream							
SSAD [8]	24.6	35.0	43.0	47.8	50.1	RGB + OF	CNN + 3D CNN
SS-TAD [9]	29.2	-	-	-	-	RGB	3D CNN
Proposal then Classification							
LearSubmission [43]	14.4	20.8	27.0	33.6	33.6	RGB	SIFT [44] + Color + CNN
SparseLearning [45]	13.5	18.2	25.7	32.9	36.1	RGB	STIP [46]
SCNN [3]	19	28.7	36.3	43.5	47.7	RGB	CNN
Daps [26]	13.9	-	-	-	-	RGB	3D CNN
SST [27]	23	-	-	-	-	RGB	3D CNN
SelfAdaptive [47]	27.7	-	-	-	-	RGB	3D CNN
R-C3D [25]	28.9	35.6	44.8	51.5	54.5	RGB	3D CNN
Two-stream RNN [48]	18.8	28.9	36.9	42.9	46.1	RGB + OF	CNN
Multi-task Learning [4]	19	28.9	36.2	43.6	47.7	RGB	3D CNN
TURN [18]	25.6	34.9	44.1	50.9	54	GRB + OF	CNN
Cascade [49]	31	41.3	50.1	56.7	60.1	RGB + OF	CNN
WeakSupervision [50]	13.7	21.1	28.2	37.7	44.4	RGB + OF	CNN
Generalization [51]	28.2	39.8	48.7	57.7	64.1	RGB	CNN
Our method	29.6	40.1	48.9	57.3	64.7	RGB + OF	3D CNN

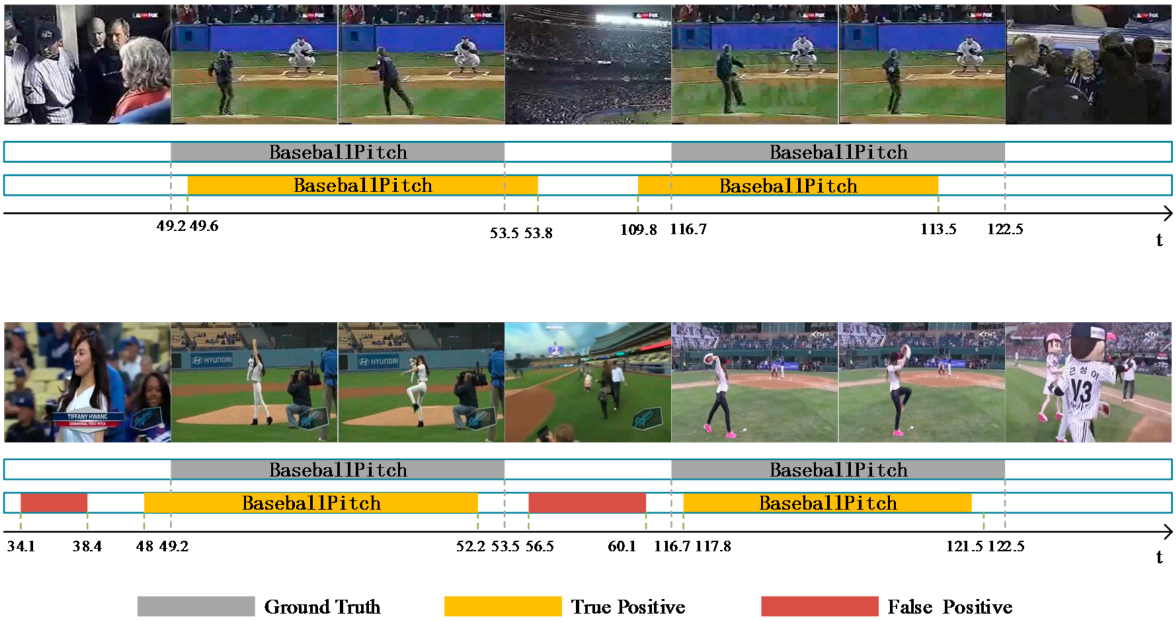


Figure 8. Detection results of the testing videos video_test_0001038 and video_test_0000324 in THUMOS'14.

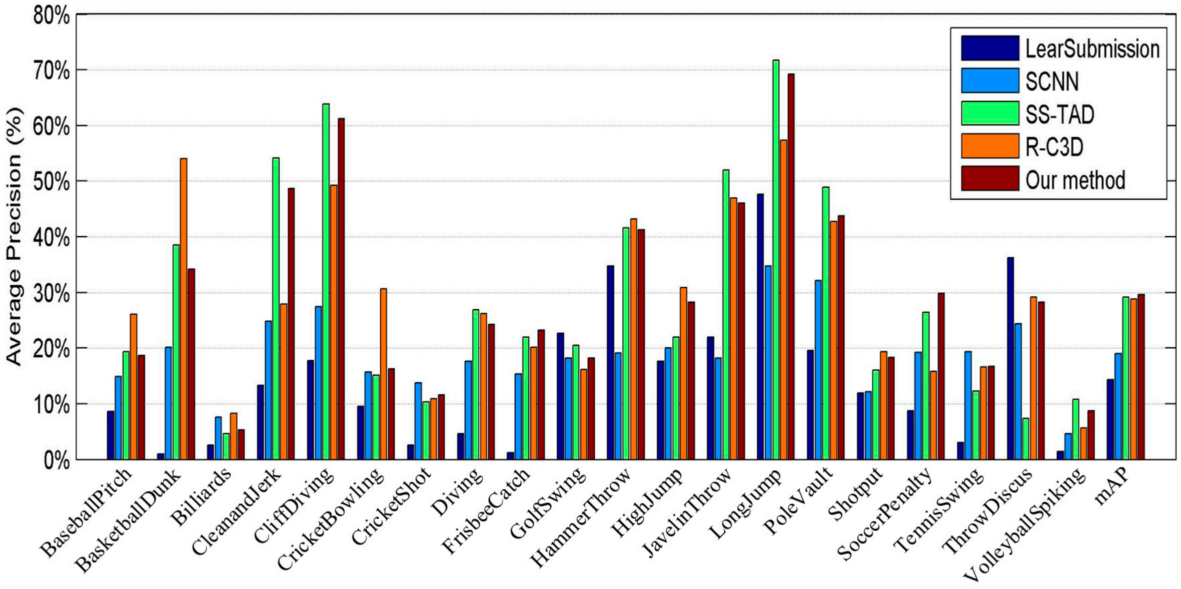


Figure 9. Detection Average Precision (AP) for each action categories with 0.5 tIoU on THUMOS'14.

Additionally, we perform the evaluation on ActivityNet1.3 dataset, whcih defines a 5-level hierarchy of action categories. Nodes on higher level represent more abstract action categories. For example, the node "Playing sports" on level-4 has child nodes "Archery", "Baton twirling", etc. on level-5. From the hierarchical action categories definition, a subset can be formed by including all action categories that belong to a certain node. Due to the large scale, it is difficult to evaluate action detection methods on this datasets. Same with the study [18], we also conduct the experiment on a subset of ActivitNet1.3. We select the "Playing sports" node on level-4 from the categories hierarchy, which has 26 action categories. Table 5 show AP for each category and mAP of the subtset 'Playing sports'. We present the performance of the state-of-the-art method on ActivityNet in Table 6. Besides recognition result, foe each method, we also specify the version of ActivityNet, and the involved subsets. Although, the compared methods are evaluated on different ActivityNet versions, and some of them and our method are evaluated on a subset of ActivityNet, the results show that our method gains a comparable performance to the ActivityNet dataset.

Table 5. Detection AP for each category and mAP of the subset 'Playing sport' of ActivityNet1.3 with 0.5 tIoU.

Archery	31.2	Doing a powerbomb	33.5	Playing kickball	47.2
Baton twirling	23.6	Doing motocross	51.6	Pole vault	38.2
Bungee jumping	46.8	Hammer throw	32.7	Powerbocking	29.5
Camel ride	48.2	High jump	33.4	Rollerblading	23.8
Cricket	37.9	Hurling	35.2	Shot put	32.1
Croquet	43.1	Javelin throw	34.8	Skateboarding	41.2
Curling	29.3	Long jump	56.7	Starting a campfire	39.2
Discus throw	44.1	Longboarding	44.3	Triple jump	27.9
Dodgeball	36.9	Paintball	43.1	**mAP**	**37.9**

Table 6. Comparison with state-of-the-art methods on ActivityNet: mAP (%) with 0.5 tIoU.

ActivityNet [41]	9.7	ActivityNet1.3
R-C3D [25]	28.4	ActivityNet1.3
Generalization [51]	40.7	ActivityNet1.3
TURN [18]	37.1	'Participating in Sports, Exercise, or Recreat.' subset of ActivityNet1.1
TURN [18]	41.2	'Housework' subset of ActivityNet1.1
Out method	37.9	'Playing sports' subset of AcvitityNet1.3

4.5. Qualitative Aanalysis of A Limitation

Although our method achieves competitive performance, it is observed that there is poor performance of our method in action cases that are with a long action duration and have an 'action pausing', such as two cases in THUMOS'14: the second action instance in video_test_00000058 and the action in video_test_0000504. Taking the former as an example, which lasts from 27.3 to 113.9 s. But, by our method, all of the detected actions are false positive and do not overlapped with the temporal region from 65.5 to 96.5 s. We present the analysis in Figure 10. In the temporal region from 65.5 to 96.5 s, because the human is almost stationary, the action is 'pausing' with less motion information. The Res3D in our method is a typical 3D CNN architecture, which is used to extract motion information of actions. Thus, it would predict the segments of this 'action pausing' temporal region into backgrounds. As illustrated in Figure 10, Window 1, which has a small percentage of predicted action segments will be removed in the discriminative temporal selective search. Window 2 has a lower score due to the overlap with 'action pausing' and Window 3 has a higher score. The Window 2 will be removed and Window 3 will be kept in NMS. However, the remaining Window 3 that has a tIoU smaller than 0.5 with the ground truth is a false positive. To address the problem of 'action pausing', the apparent information of action should be involved as the complementation of the motion information in the segment-level and window-level classifications.

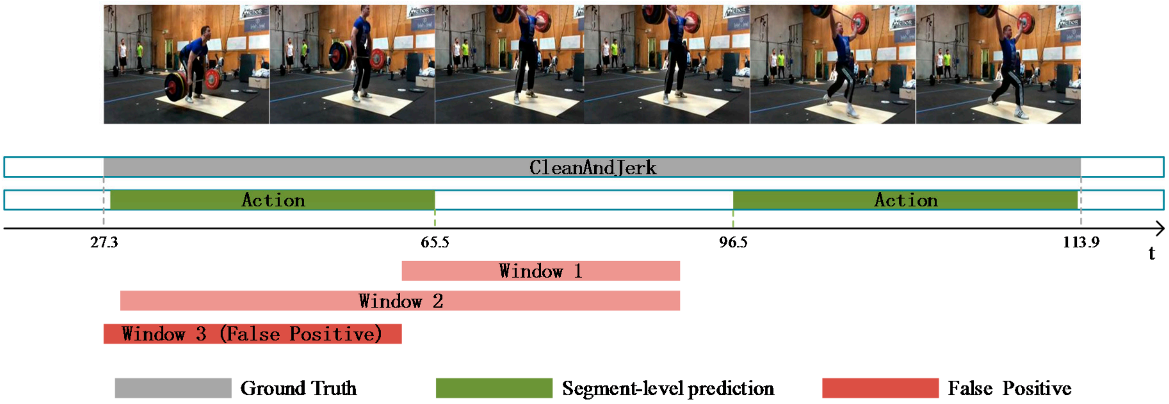


Figure 10. Result analysis of our method for the long action with an 'action pausing'.

5. Conclusions

In this paper, we propose to detect action in video from fine to coarse granularity, which is in line with people's detection habits. We build our method in the 'proposal then classification' framework. In each of the proposal and classification steps, we perform a classification from segment to window level. The proposed method has capability to localize the precise temporal boundary of action instance and considers the dependence between the segments of on action instance. Experimental results show that our proposed method achieves comparable performance with the state-of-the-art recognition performance, which only perform fine or coarse granularity classification. The universality of our method are also be demonstrated by the experimental results, which show that our proposed method has a relatively balanced performance for different action categories.

As described in Section 4.5, in the future, we should employ additional apparent information of action in the segment-level and window-level classifications to resolve the problem of 'action pausing' in the long action. On the other hand, as a typical 'proposal then classification' method, our method separates training of the proposal and classification steps and it requires two passes through for a testing video. In future work, we should implement the action detection from fine to coarse granularity in the efficient end-to-end trainable 'single stream' framework by offering a tighter integration of proposal and classification steps.

Author Contributions: G.Y. conceived the idea of this paper and wrote this paper. G.Y. and X.L. performed the experiments and analyzed the experimental results. T.L. prepared the training samples. P.J. hosted the project of CNN-based action recognition and detection.

Funding: This research was funded by Youth Innovation Promotion Association, Chinese Academy of Sciences. Grant No. 2016336.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yuan, J.; Ni, B.; Yang, X.; Kassim, A.A. Temporal action localization with pyramid of score distribution features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 3093–3102.
2. Montes, A.; Salvador, A.; Pascual, S.; Giro-i-Nieto, X. Temporal activity detection in untrimmed videos with recurrent neural networks. *arXiv*, 2017; arXiv:1608.08128.
3. Shou, Z.; Wang, D.; Chang, S. Temporal action localization in untrimmed videos via multi-stage CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1049–1058.
4. Zhu, Y.; Newsam, S. Efficient action detection in untrimmed videos via multi-task learning. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Santa Rosa, CA, USA, 24–31 March 2017; pp. 197–206.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
6. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
8. Lin, T.; Zhao, X.; Shou, Z. Single shot temporal action detection. In Proceedings of the ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017.
9. Buch, S.; Escorcia, V.; Ghanem, B.; Li, F.; Niebles, C.J. End-to-end, single-stream temporal action detection in untrimmed videos. In Proceedings of the British Machine Vision Conference, London, UK, 4–7 September 2017.
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
11. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HW, USA, 21–26 July 2017; pp. 6517–6525.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multiBox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 Octorber 2016; pp. 21–37.
13. Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; Chang, S. 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, Honolulu, HW, USA, 21–26 July 2017; pp. 1417–1426.
14. Yang, K.; Qiao, P.; Li, D.; Lv, S.; Dou, Y. Exploring temporal preservation networks for precise temporal action localization. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
15. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
16. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
17. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

18. Gao, J.; Yang, Z.; Sun, C.; Chen, K.; Nevatia, R. TURN TAP: Temporal unit regression network for temporal action proposals. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3648–3656.
19. Tran, D.; Ray, J.; Shou, Z.; Chang, S.; Paluri, M. ConvNet architecture search for spatiotemporal feature learning. *arXiv*, 2017; arXiv:1708.05038.
20. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
21. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4489–4497.
22. Wang, L.; Qiao, Y.; Tang, X. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognit. Chall.* **2014**, *1*, 2.
23. Perronnin, F.; Dance, C. Fisher kernels on visual vocabularies for image Categorization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007.
24. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3551–3558.
25. Xu, H.; Das, A.; Saenko, K. R-C3D: Region convolutional 3D network for temporal activity detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5794–5803.
26. Escorcia, V.; Heilbron, C.F.; Niebles, C.J.; Ghanem, B. DAPs: Deep action proposals for action understanding. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 Octorber 2016.
27. Buch, S.; Escorcia, V.; Shen, C.; Ghanem, B.; Niebles, C.J. SST: Single-stream temporal action proposals. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HW, USA, 21–26 July 2017; pp. 6373–6382.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. In Proceedings of the Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
30. Zeiler, M.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
31. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representation, Banff, AB, Canada, 14–16 April 2014.
32. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
33. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
34. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Li, F. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
35. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 568–576.
36. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *35*, 221–231. [CrossRef] [PubMed]
37. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y. Towards good practices for very deep two-stream convnets. *arXiv* **2015**, arXiv:1507.02159.

38. Zach, C.; Pock, T.; Bischof, H. A duality based approach for realtime tv-L1 optical flow. In Proceedings of the DAGM Symposium on Pattern Recognition, Heidelberg, Germany, 12–14 September 2007; pp. 214–223.
39. Uijlings, J.; Sande, K.; Gevers, T.; Smeulders, A. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
40. Jiang, Y.; Liu, J.; Roshan, Z.A.; Toderici, G.; Laptev, I.; Shah, M.; Sukthankar, R. THUMOS Challenge: Action Recognition with a Large Number of Classes. Available online: http://crcv.ucf.edu/THUMOS14/ (accessed on 18th August 2018).
41. Caba, H.F.; Escorcia, V.; Ghanem, B.; Niebles, J.C. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 961–970.
42. Karaman, S.; Seidenari, O.; Bimbo, D.A. Fast saliency based pooling of Fisher encoded dense trajectories. *THUMOS14 Action Recognit. Chall.* **2014**, *1*, 7.
43. Oneata, D.; Verbeek, J.; Schmid, C. The LEAR submission at Thumos 2014. 2014. Available online: https://hal.inria.fr/hal-01074442 (accessed on 9 October 2018).
44. Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
45. Caba, F.H.; Carlos, J.N.; Bernard, G. 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, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1914–1923.
46. Laptev, I. On space-time interest points. *Int. J. Comput. Vis.* **2005**, *64*, 107–123. [CrossRef]
47. Huang, J.; Li, N.; Zhang, T.; Li, G. A self-adaptive proposal model for temporal action detection based on reinforcement Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
48. Lin, T.; Zhao, X.; Fan, Z. Temporal action localization with two-stream segment-based RNN. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 3400–3404.
49. Gao, J.; Yang, Z.; Nevatia, R. Cascaded boundary regression for temporal action detection. In Proceedings of the British Machine Vision Conference, London, UK, 4–7 September 2017.
50. Wang, L.; Xiong, Y.; Lin, D.; Gool, V.L. UntrimmedNets for weakly supervised action recognition and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HW, USA, 21–26 July 2017; pp. 6402–6411.
51. Xiong, Y.; Zhao, Y.; Wang, L.; Lin, D.; Tang, X. A pursuit of temporal accuracy in general activity detection. *arXiv* **2017**, arXiv:1703.02716.