R-C3D: Region Convolutional 3D Network for Temporal Activity Detection

Huijuan Xu Abir Das Kate Saenko
Boston University
Boston, MA
{hxu, dasabir, saenko}@bu.edu

Abstract

We address the problem of activity detection in continuous, untrimmed video streams. This is a difficult task that requires extracting meaningful spatio-temporal features to capture activities, accurately localizing the start and end times of each activity. We introduce a new model, Region Convolutional 3D Network (R-C3D), which encodes the video streams using a three-dimensional fully convolutional network, then generates candidate temporal regions containing activities, and finally classifies selected regions into specific activities. Computation is saved due to the sharing of convolutional features between the proposal and the classification pipelines. The entire model is trained end-to-end with jointly optimized localization and classification losses. R-C3D is faster than existing methods (569 frames per second on a single Titan X Maxwell GPU) and achieves state-of-the-art results on THUMOS'14. We further demonstrate that our model is a general activity detection framework that does not rely on assumptions about particular dataset properties by evaluating our approach on ActivityNet and Charades. Our code is available at http://ai.bu.edu/r-c3d/

1. Introduction

Activity detection in continuous videos is a challenging problem that requires not only recognizing, but also precisely localizing activities in time. Existing state-of-the-art approaches address this task as *detection by classification*, *i.e.* classifying temporal segments generated in the form of sliding windows [13, 20, 24, 37] or via an external "proposal" generation mechanism [10, 35]. These approaches suffer from one or more of the following major drawbacks: they do not learn deep representations in an end-to-end fashion, but rather use hand-crafted features [33, 34], or deep features like VGG [28], ResNet [8], C3D [32] *etc.*, learned separately on image/video classification tasks. Such off-the-shelf representations may not be optimal for localizing activities in diverse video domains, resulting in inferior performance. Furthermore, current methods' dependence on external proposal generation or exhaustive sliding windows leads to poor computational efficiency. Finally, the sliding-window models cannot easily predict flexible activity boundaries.

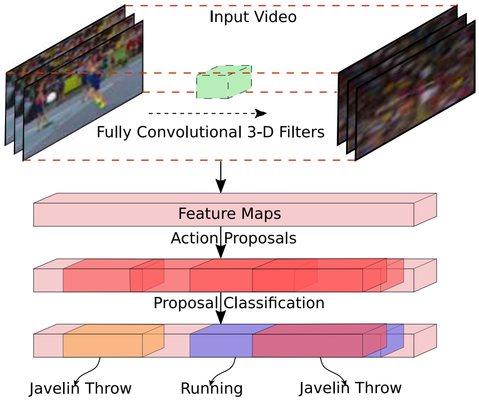

Figure 1. We propose a fast end-to-end *Region Convolutional 3D Network (R-C3D)* for activity detection in continuous video streams. The network encodes the frames with fully-convolutional 3D filters, proposes activity segments, then classifies and refines them based on pooled features within their boundaries. Our model improves both speed and accuracy compared to existing methods.

In this paper, we propose an activity detection model that addresses all of the above issues. Our *Region Convolutional 3D Network (R-C3D)* is end-to-end trainable and learns task-dependent convolutional features by jointly optimizing proposal generation and activity classification. Inspired by the Faster R-CNN [21] object detection approach, we compute fully-convolutional 3D ConvNet features and propose temporal regions likely to contain activities, then pool features within these 3D regions to predict activity classes (Figure 1). The proposal generation stage filters out many background segments and results in superior computational efficiency compared to sliding window models. Furthermore, proposals are predicted with respect to predefined anchor segments and can be of arbitrary length, allowing detection of flexible activity boundaries.

Convolutional Neural Network (CNN) features learned end-to-end have been successfully used for activity recognition [14, 27], particularly in 3D ConvNets (C3D [32]), which learn to capture spatio-temporal features. However, unlike the traditional usage of 3D ConvNets [32] where the input is short 16-frame video chunks, our method applies full convolution along the temporal dimension to encode as many frames as the GPU memory allows. Thus, rich spatio-temporal features are automatically learned from longer videos. These feature maps are shared between the activity proposal and classification subnets to save computation time and jointly optimize features for both tasks.

Alternative activity detection approaches [4, 17, 18, 29, 39] use a recurrent neural network (RNN) to encode a sequence of frame or video chunk features (*e.g.* VGG [28], C3D [32]) and predict the activity label at each time step. However, these RNN methods can only model temporal features at a fixed granularity (e.g. per-frame CNN features or 16-frame C3D features). In order to use the same classification network to classify variable length proposals into specific activities, we extend 2D region of interest (RoI) pooling to 3D which extracts a fixed-length feature representation for these proposals. Thus, our model can utilize video features at any temporal granularity. Furthermore, some RNN-based detectors rely on direct regression to predict the temporal boundaries for each activity. As shown in object detection [7, 31] and semantic segmentation [2], object boundaries obtained using a regression-only framework are inferior compared to "proposal based detection".

We perform extensive comparisons of R-C3D to state-of-the-art activity detection methods using three publicly available benchmark datasets - THUMOS'14 [12], ActivityNet [9] and Charades [26]. We achieve new state-of-the-art results on THUMOS'14 and Charades, and improved results on ActivityNet when using only C3D features.

To summarize, the main contributions of our paper are:

- an end-to-end activity detection model with combined activity proposal and classification stages that can detect arbitrary length activities;
- fast detection speeds (5x faster than current methods) achieved by sharing fully-convolutional C3D features between the proposal generation and classification parts of the network;
- extensive evaluations on three diverse activity detection datasets that demonstrate the general applicability of our model.

2. Related Work

Activity Detection There is a long history of activity recognition, or classifying trimmed video clips into fixed set of categories [11, 15, 19, 27, 33, 42]. Activity *detection* also needs to predict the start and end times of the activities within untrimmed and long videos. Existing activity detection approaches are dominated by models that use sliding windows to generate segments and subsequently classify them with activity classifiers trained on multiple features [13, 20, 24, 37]. Most of these methods have stage-wise pipelines which are not trained end-to-end. Moreover, the use of exhaustive sliding windows is computationally inefficient and constrains the boundary of the detected activities to some extent.

Recently, some approaches have bypassed the need for exhaustive sliding window search to detect activities with arbitrary lengths. [4, 17, 18, 29, 39] achieve this by modeling the temporal evolution of activities using RNNs or LSTMs networks and predicting an activity label at each time step. The deep action proposal model [4] uses LSTM to encode C3D features of every 16-frame video chunk, and directly regresses and classifies activity segments without the extra proposal generation stage. Compared to this work, we avoid recurrent layers, encoding a large video buffer with a fully-convolutional 3D ConvNet, and use 3D RoI pooling to allow feature extraction at arbitrary proposal granularity, achieving significantly higher accuracy and speed. The method in [41] tries to capture motion features at multiple resolutions by proposing a Pyramid of Score Distribution Features. However their model is not end-to-end trainable and relies on handcrafted features.

Aside from supervised activity detection, a recent work [36] has addressed weakly supervised activity localization from data labeled only with video level class labels by learning attention weights on shot based or uniformly sampled proposals. The framework proposed in [22] explores the uses of a language model and an activity length model for detection. Spatio-temporal activity localization [38, 40] have also been explored to some extent. We only focus on supervised temporal activity localization.

Object Detection Activity detection in untrimmed videos is closely related to object detection in images. The inspiration for our work, Faster R-CNN [21], extends R-CNN [7] and Fast R-CNN [6] object detection approaches, incorporating RoI pooling and a region proposal network. Compared to recent object detection models *e.g.*, SSD [16] and R-FCN [3], Faster R-CNN is a general and robust object detection framework that has been deployed on different datasets with little data augmentation effort. Like Faster R-CNN, our R-C3D model is also designed with the goal of easy deployment on varied activity detection datasets. It avoids making certain assumptions based on unique characteristics of a dataset, such as the UPC model for ActivityNet [18] which assumes that each video contains a single activity class. We show the effectiveness of our model on three different types of activity detection datasets, the most extensive evaluation to our knowledge.

3. Approach

We propose a *Region Convolutional 3D Network (R-C3D)*, a novel convolutional neural network for activity detection in continuous video streams. The network, illus-

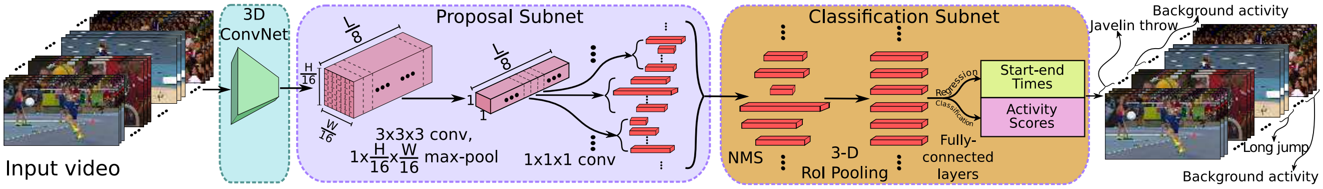

Figure 2. R-C3D model architecture. The 3D ConvNet takes raw video frames as input and computes convolutional features. These are input to the Proposal Subnet that proposes candidate activities of variable length along with confidence scores. The Classification Subnet filters the proposals, pools fixed size features and then predicts activity labels along with refined segment boundaries.

trated in Figure 2, consists of three components: a shared 3D ConvNet feature extractor [32], a temporal proposal stage, and an activity classification and refinement stage. To enable efficient computation and end-to-end training, the proposal and classification sub-networks share the same C3D feature maps. The proposal subnet predicts variable length temporal segments that potentially contain activities, while the classification subnet classifies these proposals into specific activity categories or background, and further refines the proposal segment boundaries. A key innovation is to extend the 2D RoI pooling in Faster R-CNN to 3D RoI pooling which allows our model to extract features at various resolutions for variable length proposals. Next, we describe the shared video feature hierarchies in Sec. 3.1, the temporal proposal subnet in Sec. 3.2 and the classification subnet in Sec. 3.3. Sections 3.4 and 3.5 detail the optimization strategy during training and testing respectively.

3.1. 3D Convolutional Feature Hierarchies

We use a 3D ConvNet to extract rich spatio-temporal feature hierarchies from a given input video buffer. It has been shown that both spatial and temporal features are important for representing videos, and a 3D ConvNet encodes rich spatial and temporal features in a hierarchical manner. The input to our model is a sequence of RGB video frames with dimension $\mathbb{R}^{3\times L\times H\times W}$. The architecture of the 3D ConvNet is taken from the C3D architecture proposed in [32]. However, unlike [32], the input to our model is of variable length. We adopt the convolutional layers (`conv1a` to `conv5b`) of C3D, so a feature map $C_{conv5b} \in \mathbb{R}^{512\times\frac{L}{8}\times\frac{H}{16}\times\frac{W}{16}}$ (512 is the channel dimension of the layer `conv5b`) is produced as the output of this subnetwork. We use C_{conv5b} activations as the shared input to the proposal and classification subnets. The height (H) and width (W) of the frames are taken as 112 each following [32]. The number of frames L can be arbitrary and is only limited by memory.

3.2. Temporal Proposal Subnet

To allow the model to predict variable length proposals, we incorporate anchor segments into the temporal proposal sub-network. The subnet predicts potential proposal segments with respect to anchor segments and a binary label indicating whether the predicted proposal contains an activity or not. The anchor segments are pre-defined multi-scale windows centered at $L/8$ uniformly distributed temporal locations. Each temporal location specifies K anchor segments, each at a different fixed scale. Thus, the total number of anchor segments is $(L/8) * K$. The same set of K anchor segments exists in different temporal locations, which ensures that the proposal prediction is temporally invariant. The anchors serve as reference activity segments for proposals at each temporal location, where the maximum number of scales K is dataset dependent.

To obtain features at each temporal location for predicting proposals with respect to these anchor segments, we first add a 3D convolutional filter with kernel size 3×3×3 on top of C_{conv5b} to extend the temporal receptive field for the temporal proposal subnet. Then, we downsample the spatial dimensions (from $\frac{H}{16}\times\frac{W}{16}$ to 1×1) to produce a *temporal* only feature map $C_{tpn} \in \mathbb{R}^{512\times\frac{L}{8}\times 1\times 1}$ by applying a 3D max-pooling filter with kernel size $1\times\frac{H}{16}\times\frac{W}{16}$. The 512-dimensional feature vector at each temporal location in C_{tpn} is used to predict a relative offset $\{\delta c_i, \delta l_i\}$ to the center location and the length of each anchor segment $\{c_i, l_i\}, i \in \{1, \cdots, K\}$. It also predicts the binary scores for each proposal being an activity or background. The proposal offsets and scores are predicted by adding two 1×1×1 convolutional layers on top of C_{tpn}.

Training: For training, we need to assign positive/negative labels to the anchor segments. Following the standard practice in object detection [21], we choose a positive label if the anchor segment 1) overlaps with some ground-truth activity with Intersection-over-Union (IoU) higher than 0.7, or 2) has the highest IoU overlap with some ground-truth activity. If the anchor has IoU overlap lower than 0.3 with all ground-truth activities, then it is given a negative label. All others are held out from training. For proposal regression, ground truth activity segments are transformed with respect to nearby anchor segments using the coordinate transformations described in Sec. 3.4. We sample balanced batches with a positive/negative ratio of 1 : 1.

3.3. Activity Classification Subnet

The activity classification stage has three main functions: 1) selecting proposal segments from the previous stage, 2) three-dimensional region of interest (3D RoI) pooling to extract fixed-size features for selected proposals, and 3) activity classification and boundary regression for the selected proposals based on the pooled features.

Some activity proposals generated by the proposal subnet highly overlap with each other and some have a low proposal score indicating low confidence. Following the standard practice in object detection [5, 21] and activity detection [24, 39], we employ a greedy Non-Maximum Suppression (NMS) strategy to eliminate highly overlapping and low confidence proposals. The NMS threshold is set as 0.7.

The selected proposals can be of arbitrary length. However we need to extract fixed-size features for each of them in order to use fully connected layers for further activity classification and regression. We design a 3D RoI pooling layer to extract the fixed-size volume features for each variable-length proposal from the shared convolutional features $C_{conv5b} \in \mathbb{R}^{512 \times (L/8) \times 7 \times 7}$ (shared with the temporal proposal subnet). Specifically, in 3D RoI pooling, an input feature volume of size, say, $l \times h \times w$ is divided into $l_s \times h_s \times w_s$ sub-volumes each with approximate size $\frac{l}{l_s} \times \frac{h}{h_s} \times \frac{w}{w_s}$, and then max pooling is performed inside each sub-volume. In our case, suppose a proposal has the feature volume of $l_p \times 7 \times 7$ in C_{conv5b}, then this feature volume will be divided into $1 \times 4 \times 4$ grids and max pooled inside each grid. Thus, proposals of arbitrary lengths give rise to output volume features of the same size $512 \times 1 \times 4 \times 4$.

The output of the 3D RoI pooling is fed to a series of two fully connected layers. Here, the proposals are classified to activity categories by a classification layer and the refined start-end times for these proposals are given by a regression layer. The classification and regression layers are also two separate fully connected layers and for both of them the input comes from the aforementioned fully connected layers (after the 3D RoI pooling layer).

Training: We need to assign an activity label to each proposal for training the classifier subnet. An activity label is assigned if the proposal has the highest IoU overlap with a ground-truth activity, and at the same time, the IoU overlap is greater than 0.5. A background label (no activity) is assigned to proposals with IoU overlap lower than 0.5 with all ground-truth activities. Training batches are chosen with positive/negative ratio of 1 : 3.

3.4. Optimization

We train the network by optimizing both the classification and regression tasks jointly for the two subnets. The softmax loss function is used for classification, and smooth L1 loss function [6] is used for regression. Specifically, the objective function is given by:

$$Loss = \frac{1}{N_{cls}} \sum_i L_{cls}(a_i, a_i^*) + \lambda \frac{1}{N_{reg}} \sum_i a_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where N_{cls} and N_{reg} stand for batch size and the number of anchor/proposal segments, λ is the loss trade-off parameter and is set to a value 1. i is the anchor/proposal segments index in a batch, a_i is the predicted probability of the proposal or activities, a_i^* is the ground truth, $t_i = \{\delta \hat{c}_i, \delta \hat{l}_i\}$ represents predicted relative offset to anchor segments or proposals. $t_i^* = \{\delta c_i, \delta l_i\}$ represents the coordinate transformation of ground truth segments to anchor segments or proposals. The coordinate transformations are computed as follows:

$$\begin{cases} \delta c_i = (c_i^* - c_i)/l_i \\ \delta l_i = log(l_i^*/l_i) \end{cases} \quad (2)$$

where c_i and l_i are the center location and the length of anchor segments or proposals while c_i^* and l_i^* denote the same for the ground truth activity segments.

In our R-C3D model, the above loss function is applied for both the temporal proposal subnet and the activity classification subnet. In the proposal subnet, the binary classification loss L_{cls} predicts whether the proposal contains an activity or not, and the regression loss L_{reg} optimizes the relative displacement between proposals and ground truths. In the proposal subnet the losses are activity class agnostic. For the activity classification subnet, the multiclass classification loss L_{cls} predicts the specific activity class for the proposal, and the number of classes are the number of activities plus one for the background. The regression loss L_{reg} optimizes the relative displacement between activities and ground truths. All four losses for the two subnets are optimized jointly.

3.5. Prediction

Activity prediction in R-C3D consists of two steps. First, the proposal subnet generates candidate proposals and predicts the start-end time offsets as well as proposal score for each. Then the proposals are refined via NMS with threshold value 0.7. After NMS, the selected proposals are fed to the classification network to be classified into specific activity classes, and the activity boundaries of the predicted proposals are further refined by the regression layer. The boundary prediction in both proposal subnet and classification subnet is in the form of relative displacement of center point and length of segments. In order to get the start time and end time of the predicted proposals or activities, inverse coordinate transformation to Equation 2 is performed.

R-C3D accepts variable length input videos. However, to take advantage of the vectorized implementation in fast deep learning libraries, we pad the last few frames of short videos with last frame, and break long videos into buffers (limited by memory only). NMS at a lower threshold (0.1 less than the mAP evaluation threshold) is applied to the predicted activities to get the final activity predictions.

Table 1. Activity detection results on THUMOS'14 (in percentage). mAP at different IoU thresholds α are reported. The top three performers on the THUMOS'14 challenge leaderboard and other results reported in existing papers are shown.

	α				
	0.1	0.2	0.3	0.4	0.5
Karaman et al. [13]	4.6	3.4	2.1	1.4	0.9
Wang et al. [37]	18.2	17.0	14.0	11.7	8.3
Oneata et al. [20]	36.6	33.6	27.0	20.8	14.4
Heilbron et al. [10]	-	-	-	-	13.5
Escorcia et al. [4]	-	-	-	-	13.9
Richard et al. [22]	39.7	35.7	30.0	23.2	15.2
Yeung et al. [39]	48.9	44.0	36.0	26.4	17.1
Yuan et al. [41]	51.4	42.6	33.6	26.1	18.8
Shou et al. [24]	47.7	43.5	36.3	28.7	19.0
Shou et al. [23]	-	-	40.1	29.4	23.3
R-C3D (our one-way buffer)	51.6	49.2	42.8	33.4	27.0
R-C3D (our two-way buffer)	**54.5**	**51.5**	**44.8**	**35.6**	**28.9**

4. Experiments

We evaluate R-C3D on three large-scale activity detection datasets - THUMOS'14 [12], Charades [26] and ActivityNet [9]. Sections 4.1, 4.2, 4.3 provide the experimental details and evaluation results on these three datasets. Results are shown in terms of mean Average Precision - mAP@α where α denotes different Intersection over Union (IoU) thresholds, as is the common practice in the literature. Section 4.4 provides the detection speed comparison with state-of-the-art methods.

4.1. Experiments on THUMOS'14

THUMOS'14 activity detection dataset contains over 24 hours of video from 20 different sport activities. The training set contains 2765 trimmed videos while the validation and the test sets contain 200 and 213 untrimmed videos respectively. This dataset is particularly challenging as it consists of very long videos (up to a few hundreds of seconds) with multiple activity instances of very small duration (up to few tens of seconds). Most videos contain multiple activity instances of the same activity class. In addition, some videos contain activity segments from different classes.

Experimental Setup: We divide 200 untrimmed videos from the validation set into 180 training and 20 held out videos to get the best hyperparameter setting. All 200 videos are used as the training set and the final results are reported on 213 test videos. Since the GPU memory is limited, we first create a buffer of 768 frames at 25 frames per second (fps) which means approximately 30 seconds of video. Our choice is motivated by the fact that 99.5% of all activity segments in the validation set (used here as the training set) are less than 30 seconds long. These buffers of frames act as inputs to R-C3D . We can create the buffer by sliding from the beginning of the video to the end, denoted as the 'one-way buffer'. An additional pass from the end of the video to the beginning is used to increase the amount of training data, denoted as the 'two-way buffer'. We initialize the 3D ConvNet part of our model with C3D weights trained on Sports-1M and finetuned on UCF101 released by the authors in [32]. We allow all the layers of R-C3D to be trained on THUMOS'14 with a fixed learning rate of 0.0001.

Table 2. Per-class AP at IoU threshold $\alpha = 0.5$ on THUMOS'14 (in percentage).

	[20]	[39]	[24]	R-C3D (ours)
Baseball Pitch	8.6	14.6	14.9	**26.1**
Basketball Dunk	1.0	6.3	20.1	**54.0**
Billiards	2.6	**9.4**	7.6	8.3
Clean and Jerk	13.3	**42.8**	24.8	27.9
Cliff Diving	17.7	15.6	27.5	**49.2**
Cricket Bowling	9.5	10.8	15.7	**30.6**
Cricket Shot	2.6	3.5	**13.8**	10.9
Diving	4.6	10.8	17.6	**26.2**
Frisbee Catch	1.2	10.4	15.3	**20.1**
Golf Swing	**22.6**	13.8	18.2	16.1
Hammer Throw	34.7	28.9	19.1	**43.2**
High Jump	17.6	**33.3**	20.0	30.9
Javelin Throw	22.0	20.4	18.2	**47.0**
Long Jump	47.6	39.0	34.8	**57.4**
Pole Vault	19.6	16.3	32.1	**42.7**
Shotput	11.9	16.6	12.1	**19.4**
Soccer Penalty	8.7	8.3	**19.2**	15.8
Tennis Swing	3.0	5.6	**19.3**	16.6
Throw Discus	**36.2**	29.5	24.4	29.2
Volleyball Spiking	1.4	5.2	4.6	**5.6**
mAP@0.5	14.4	17.1	19.0	**28.9**

The number of anchor segments K chosen for this dataset is 10 with specific scale values [2, 4, 5, 6, 8, 9, 10, 12, 14, 16]. The values are chosen according to the distribution of the activity durations in the training set. At 25 fps and temporal pooling factor of 8 (C_{tpn} downsamples the input by 8 temporally), the anchor segments correspond to segments of duration between 0.64 and 5.12 seconds[1]. Note that, the predicted proposals or activities are relative to the anchor segments but not limited to the anchor boundaries, enabling our model to detect variable-length activities.

Results: As a sanity check, we first evaluate the performance of the temporal proposal subnet.A predicted proposal is marked correct if its IoU with a ground truth activity is more than 0.7, otherwise it is considered incorrect. With this binary setting, precision and recall values of the temporal proposal subnet are 85% and 83% respectively.

In Table 1, we present a comparative evaluation of the activity detection performance of R-C3D with existing state-of-the-art approaches in terms of mAP at IoU thresholds 0.1-0.5 (denoted as α). For both the one-way buffer setting and the two-way buffer setting we achieve new state-of-the-art for all five α values. In the one-way setting, mAP@0.5 is 27.0% which is an 3.7% absolute improvement from the state-of-the-art. The two-way buffer setting further

[1] $2 * 8/25 = 0.64$ and $16 * 8/25 = 5.12$

increases the mAP values at all the IoU thresholds with mAP@0.5 reaching as far as 28.9%. Our model comprehensively outperforms the current state-of-the-art by a large margin (28.9% compared to 23.3% as reported in [23]).

The Average Precision (AP) for each class in THUMOS'14 at IoU threshold 0.5 for the two-way buffer setting is shown in Table 2. R-C3D outperforms the all the methods in most classes and shows significant improvement (by more than 20% absolute AP over the next best) for activities *e.g.*, Basketball Dunk, Cliff Diving, and Javelin Throw. For some of the activities, our method is only second to the best performing ones by a very small margin (*e.g.*, Billiards or Cricket Shot). Figure 3(a) shows some representative qualitative results from two videos in this dataset.

4.2. Experiments on ActivityNet

The ActivityNet [9] dataset consists of untrimmed videos and is released in three versions. We use the latest release (1.3) which has 10024, 4926 and 5044 videos containing 200 different types of activities in the train, validation and test sets respectively. Most videos contain activity instances of a single class covering a great deal of the video. Compared to THUMOS'14, this is a large-scale dataset both in terms of the number of activities involved and the amount of video. Researchers have taken part in the ActivityNet challenge [1] held on this dataset. The performances of the participating teams are evaluated on test videos for which the ground truth annotations are not public. In addition to evaluating on the validation set, we show our performance on the test set after evaluating it on the challenge server.

Experimental Setup: Similar to THUMOS'14, the length of the input buffer is set to 768 but, as the videos are long, we sample frames at 3 fps to fit it in the GPU memory. This makes the duration of the buffer approximately 256 seconds covering over 99.99% training activities. The considerably long activity durations prompt us to set the number of anchor segments K to be as high as 20. Specifically, we chose the following scales - [1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32, 40, 48, 56, 64]. Thus the shortest and the longest anchor segments are of durations 2.7 and 170 seconds respectively covering 95.6% training activities.

Considering the vast domain difference of the activities between Sports-1M and ActivityNet, we finetune the Sports-1M pretrained 3D ConvNet model [32] with the training videos of ActivityNet. We initialize the 3D ConvNet with these finetuned weights. AcitivityNet being a large scale dataset, the training takes more epochs. As a speed-efficiency trade-off, we freeze the first two convolutional layers in our model during training. The learning rate is kept fixed at 10^{-4} for first 10 epochs and is decreased to 10^{-5} for the last 5 epochs. Based on the improved results on the THUMOS'14, we choose the two-way buffer setting with horizontal flipping of frames for data augmentation.

Table 3. Detection results on ActivityNet in terms of mAP@0.5 (in percentage). The top half of the table shows performance from methods using additional handcrafted features while the bottom half shows approaches using deep features only (including ours). Results for [29] are taken from [1]

	train data	validation	test
G. Singh *et. al.* [30]	train	34.5	36.4
B. Singh *et. al.* [29]	train+val	-	28.8
UPC [18]	train	22.5	22.3
R-C3D (ours)	train	**26.8**	**26.8**
R-C3D (ours)	train+val	-	**28.4**

Results: In Table 3 we show the performance of R-C3D and compare with existing published approaches. Results are shown for two different settings. In the first setting, only the training set is used for training and the performance is shown for either the validation or test data or both. In the second setting, training is done on both training and validation sets while the performance is shown on the test set. The table shows that the proposed method does achieve a performance better than methods not using handcrafted features *e.g.*, UPC [18]. UPC is the most fair comparison as it also uses only C3D features. However, it relies on a strong assumption that each video in ActivityNet just contains one activity class. Our approach obtains an improvement of 4.3% on the validation set and 4.5% on the test set over UPC [18] in terms of mAP@0.5 without any such strong assumptions. When both training and validation sets are used for training, the performance improves further by 1.6%. The ActivityNet Challenge in 2017 introduced a new evaluation metric where mAP at 10 evenly distributed thresholds between 0.5 and 0.95 are averaged to get the *average mAP*. Using only training data to train R-C3D, the average mAP for the validation and test set are 12.7% and 13.1% respectively. On the other hand, if both training and validation data is used during training, the average mAP for the test set increases to 16.7% showing the benefit of our end-to-end model when more data is available for training.

R-C3D falls slightly behind [29] which uses LSTM based tracking and performs activity prediction using deep features as well as optical flow features from the tracked trajectories. The approach in [30] also uses handcrafted motion features like MBH on top of inception and C3D features in addition to dynamic programing based post processing. However, the heavy use of an ensemble of hand-engineered features and dataset dependent heuristics not only stops these methods from learning in an end-to-end fashion but makes them less general across datasets. Unlike these methods, R-C3D is trainable completely end-to-end and is easily extensible to other datasets with little parameter tuning, providing better generalization performance. Our method is also capable of using hand engineered features with a possible boost to performance, and we keep

Table 4. Activity detection results on Charades (in percentage). We report the results using the same evaluation metric as in [25].

	mAP	
	standard	post-process
Random [25]	4.2	4.2
RGB [25]	7.7	8.8
Two-Stream [25]	7.7	10.0
Two-Stream+LSTM [25]	8.3	8.8
Sigurdsson et al. [25]	9.6	12.1
R-C3D (ours)	**12.4**	**12.7**

this as a future task. Figure 3(b) shows some representative qualitative results from this dataset.

4.3. Experiments on Charades

Charades [26] is a recently introduced dataset for activity classification and detection. The activity detection task involves daily life activities from 157 classes. The dataset consists of 7985 train and 1863 test videos. The videos are recorded by Amazon Mechanical Turk users based on provided scripts. Apart from low illumination, diversity and casual nature of the videos containing day-to-day activities, an additional challenge of this dataset is the abundance of overlapping activities, sometimes multiple activities having exactly the same start and end times (typical examples include pairs of activities like 'holding a phone' and 'playing with a phone' or 'holding a towel' and 'tidying up a towel').

Experimental Setup: For this dataset we sample frames at 5 fps, and the input buffer is set to contain 768 frames. This makes the duration of the buffer approximately 154 seconds covering all the ground truth activity segments in Charades train set. As the activity segments for this dataset are longer, we choose the number of anchor segments K to be 18 with specific scale values [1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32, 40, 48]. So the shortest anchor segment has a duration of 1.6 seconds and the longest anchor segment has a duration of 76.8 seconds. Over 99.96% of the activities in the training set is under 76.8 seconds. For this dataset we, additionally, explored slightly different settings of the anchor segment scales, but found that our model is not very sensitive to this hyperparameter.

We first finetune the Sports-1M pretrained C3D model [32] on the Charades training set at the same 5 fps and initialize the 3D ConvNet part of our model with these finetuned weights. Next, we train R-C3D end-to-end on Charades by freezing the first two convolutional layers in order to accelerate training. The learning rate is kept fixed at 0.0001 for the first 10 epochs and then decreased to 0.00001 for 5 further epochs. We augment the data by following the two-way buffer setting and horizontal flipping of frames.

Results: Table 4 provides a comparative evaluation of the proposed model with various baseline models reported in [25]. This approach [25] trains a CRF based video classification model (asynchronous temporal fields) and evaluates the prediction performance on 25 equidistant frames by making a multi-label prediction for each of these frames. The activity localization result is reported in terms of mAP metric on these frames. For a fair comparison, we map our activity segment prediction to 25 equidistant frames and evaluate using the same mAP evaluation metric. A second evaluation strategy proposed in this work relies on a post-processing stage where the frame level predictions are averaged across 20 frames leading to more spatial consistency. As shown in the Table 4, our model outperforms the asynchronous temporal fields model proposed in [25] as well as the different baselines reported in the same paper. While the improvement over the standard method is as high as 2.8%, the improvement after the post-processing is not as high. One possible reason could be that our end-to-end fully convolutional model captures the spatial consistency implicitly without requiring any manually-designed postprocessing.

Following the standard practice we also evaluated our model in terms of mAP@0.5 which comes out to be 9.3%. The performance is not at par with other datasets presumably because of the inherent challenges involved in Charades *e.g.*, the low illumination indoor scenes or the multi-label nature of the data. Initialization with a better C3D classification model trained on indoor videos with these challenging conditions may further boost the performance. Figure 3(c) shows some representative qualitative results from one video in this dataset.

One of the major challenges of this dataset is the presence of a large number of temporally overlapping activities. The results show that our model is capable of handling such scenarios. This is achieved by the ability of the proposal subnet to produce possibly overlapping activity proposals and is further facilitated by region offset regression.

Table 5. Activity detection speed during inference.

	FPS
S-CNN [24]	60
DAP [4]	134.1
R-C3D (ours on Titan X Maxwell)	**569**
R-C3D (ours on Titan X Pascal)	**1030**

4.4. Activity Detection Speed

In this section, we compare detection speed of our model with two other state-of-the-art methods. The comparison results are shown in Table 5. S-CNN [24] uses a time-consuming sliding window strategy and predicts at 60 fps. DAP [4] incorporates a proposal prediction step on top of LSTM and predicts at 134.1 fps. R-C3D constructs the proposal and classification pipeline in an end-to-end fashion and these two stages share the features making it significantly faster. The speed of execution is 569 fps on a single Titan-X (Maxwell) GPU for the proposal and classification stages together. On the upgraded Titan-X (Pascal) GPU, our inference speed reaches even higher (1030 fps). One of the reasons of the speedup of R-C3D over DAP may come from

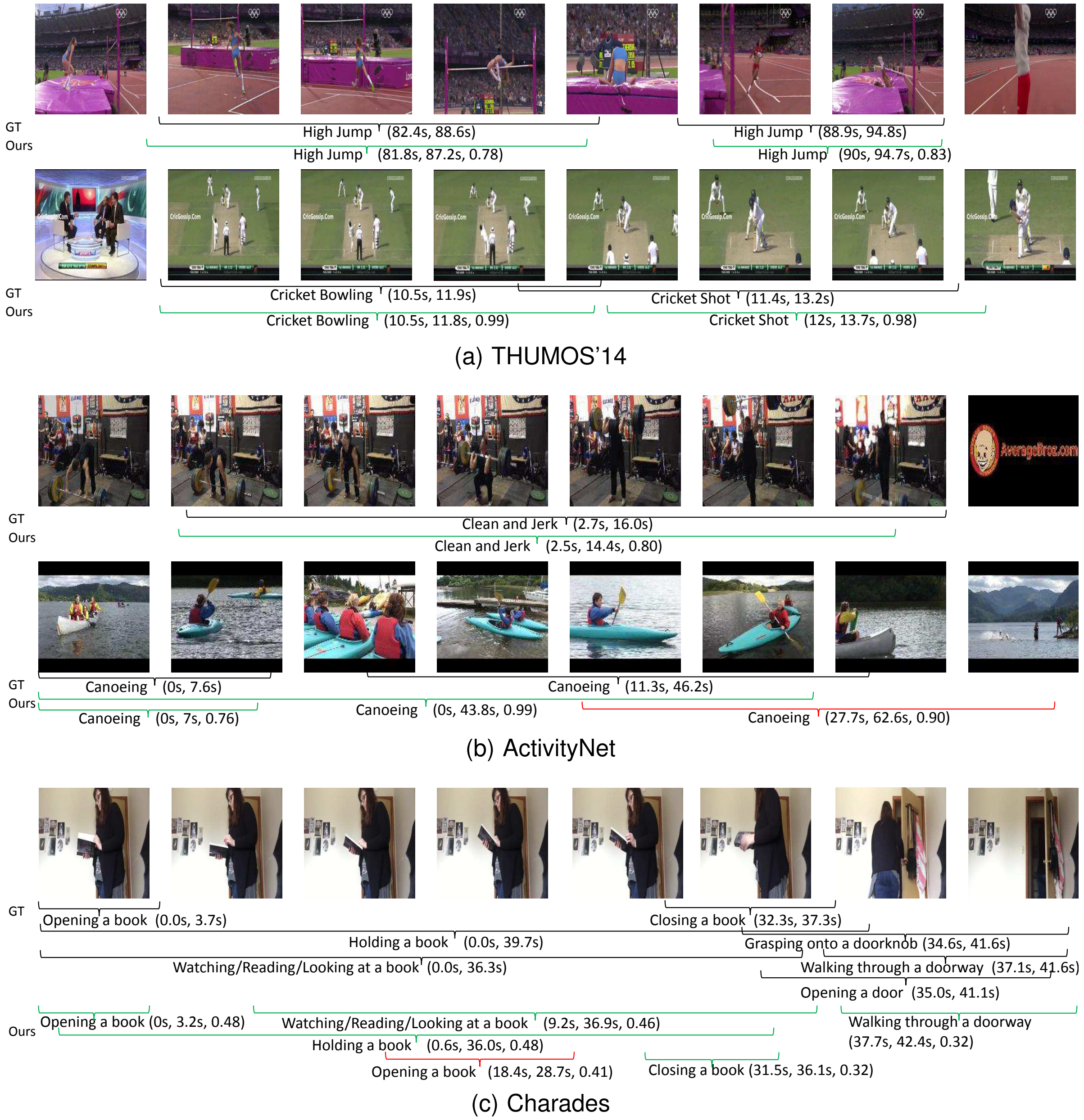
GT
Ours
High Jump (82.4s, 88.6s)
High Jump (88.9s, 94.8s)
High Jump (81.8s, 87.2s, 0.78)
High Jump (90s, 94.7s, 0.83)
GT
Ours
Cricket Bowling (10.5s, 11.9s)
Cricket Shot (11.4s, 13.2s)
Cricket Bowling (10.5s, 11.8s, 0.99)
Cricket Shot (12s, 13.7s, 0.98)
(a) THUMOS'14
GT
Ours
Clean and Jerk (2.7s, 16.0s)
Clean and Jerk (2.5s, 14.4s, 0.80)
GT
Ours
Canoeing (0s, 7.6s)
Canoeing (11.3s, 46.2s)
Canoeing (0s, 43.8s, 0.99)
Canoeing (0s, 7s, 0.76)
Canoeing (27.7s, 62.6s, 0.90)
(b) ActivityNet
GT
Opening a book (0.0s, 3.7s)
Closing a book (32.3s, 37.3s)
Holding a book (0.0s, 39.7s)
Grasping onto a doorknob (34.6s, 41.6s)
Watching/Reading/Looking at a book (0.0s, 36.3s)
Walking through a doorway (37.1s, 41.6s)
Opening a door (35.0s, 41.1s)
Ours
Opening a book (0s, 3.2s, 0.48)
Watching/Reading/Looking at a book (9.2s, 36.9s, 0.46)
Walking through a doorway (37.7s, 42.4s, 0.32)
Holding a book (0.6s, 36.0s, 0.48)
Opening a book (18.4s, 28.7s, 0.41)
Closing a book (31.5s, 36.1s, 0.32)
(c) Charades

Figure 3. Qualitative visualization of the predicted activities by R-C3D (best viewed in color). Figure (a) and (b) show results for two videos each in THUMOS'14 and ActivityNet. (c) shows the result for one video from Charades. Groundtruth activity segments are marked in black. Predicted activity segments are marked in green for correct predictions and in red for wrong ones. Predicted activities with IoU more than 0.5 are considered as correct. Corresponding start-end times and confidence score are shown inside brackets.

the fact that the LSTM recurrent architecture in DAP takes time to unroll, while R-C3D directly accepts a wide range of frames as input and the convolutional features are shared by the proposal and classification subnets.

5. Conclusion

We introduce R-C3D, the first end-to-end temporal proposal classification network for activity detection. We evaluate our approach on three large-scale data sets with very diverse characteristics, and demonstrate that it can detect activities faster and more accurately than existing models based on 3D Convnets. Additional features can be incorporated into R-C3D to further boost the activity detection result. One future direction may be to integrate R-C3D with hand-engineered motion features for improved activity prediction without sacrificing speed.

Acknowledgement: We would thank Lu He for the meaningful discussions. This research was supported by the NSF IIS-1212928 grant, the National Geospatial Agency, and a hardware grant from NVIDIA.

References

[1] ActivitNet Large Scale Activity Recognition Challenge. http://activity-net.org/challenges/2016/data/anet_challenge_summary.pdf, 2016. 6

[2] J. Carreira and C. Sminchisescu. CPMC: Automatic Object Segmentation using Constrained Parametric Min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012. 2

[3] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *Neural Information Processing Systems*, pages 379–387, 2016. 2

[4] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. DAPs: Deep Action Proposals for Action Understanding. In *European Conference on Computer Vision*, pages 768–784, 2016. 2, 5, 7

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010. 4

[6] R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2, 4

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 2

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[9] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 2, 5, 6

[10] F. C. Heilbron, J. C. Niebles, and B. Ghanem. Fast Temporal Activity Proposals for Efficient Detection of Human Actions in Untrimmed Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1914–1923, 2016. 1, 5

[11] S. Ji, W. Xu, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:221–231, 2013. 2

[12] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS Challenge: Action Recognition with a Large Number of Classes. http://crcv.ucf.edu/THUMOS14/, 2014. 2, 5

[13] S. Karaman, L. Seidenari, and A. D. Bimbo. Fast Saliency Based Pooling of Fisher Encoded Dense Trajectories. In *ECCV THUMOS Workshop*, 2014. 1, 2, 5

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*, pages 21–37, 2016. 2

[17] S. Ma, L. Sigal, and S. Sclaroff. Learning Activity Progression in LSTMs for Activity Detection and Early Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950, 2016. 2

[18] A. Montes, A. Salvador, and X. G. i Nieto. Temporal Activity Detection in Untrimmed Videos with Recurrent Neural Networks. *arXiv preprint arXiv:1608.08128*, 2016. 2, 6

[19] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond Short Snippets: Deep Networks for Video Classification. In *IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015. 2

[20] D. Oneata, J. Verbeek, and C. Schmid. The LEAR submission at Thumos 2014. *ECCV THUMOS Workshop*, 2014. 1, 2, 5

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Neural Information Processing Systems*, pages 91–99, 2015. 1, 2, 3, 4

[22] A. Richard and J. Gall. Temporal Action Detection Using a Statistical Language Model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[23] Z. Shou, J. Chan, A. Zareian, K. Miyazaway, and S.-F. Chang. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5, 6

[24] Z. Shou, D. Wang, and S.-F. Chang. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 4, 5, 7

[25] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous Temporal Fields for Action Recognition. *arXiv preprint arXiv:1612.06371*, 2017. 7

[26] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision*, 2016. 2, 5, 7

[27] K. Simonyan and A. Zisserman. Two-stream Convolutional Networks for Action Recognition in Videos. In *Neural Information Processing Systems*, pages 568–576, 2014. 1, 2

[28] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. In *IEEE Conference on Learning Representations*, 2015. 1, 2

[29] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 6

[30] G. Singh and F. Cuzzolin. Untrimmed Video Classification for Activity Detection: submission to ActivityNet Challenge. *arXiv preprint arXiv:1607.01979*, 2016. 6

[31] C. Szegedy, A. Toshev, and D. Erhan. Deep Neural Networks for Object Detection. In *Neural Information Processing Systems*, pages 2553–2561, 2013. 2

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 1, 2, 3, 5, 6, 7

[33] H. Wang and C. Schmid. Action Recognition with Improved Trajectories. In *IEEE International Conference on Computer Vision*, pages 3551–3558, 2013. 1, 2

[34] L. Wang, Y. Qiao, and X. Tang. Video Action Detection with Relational Dynamic-Poselets. In *European Conference on Computer Vision*, pages 565–580, 2014. 1

[35] L. Wang, Y. Qiao, X. Tang, and L. V. Gool. Actionness Estimation using Hybrid Fully Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2016. 1

[36] L. Wang, Y. Xiong, D. Lin, and L. V. Gool. UntrimmedNets for Weakly Supervised Action Recognition and Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[37] L. Wang, Y. Yu Qiao, and X. Tang. Action Recognition and Detection by Combining Motion and Appearance Features. *ECCV THUMOS Workshop*, 1, 2014. 1, 2, 5

[38] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to Track for Spatio-Temporal Action Localization. In *IEEE International Conference on Computer Vision*, 2015. 2

[39] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end Learning of Action Detection from Frame Glimpses in Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016. 2, 4, 5

[40] G. Yu and J. Yuan. Fast Action Proposals for Human Action Detection and Search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[41] J. Yuan, B. Ni, X. Yang, and A. A. Kassim. Temporal Action Localization with Pyramid of Score Distribution Features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2016. 2, 5

[42] J. Zheng, Z. Jiang, and R. Chellappa. Cross-view Action Recognition via Transferable Dictionary Learning. *IEEE Transactions on Image Processing*, 25(6):2542–2556, 2016. 2