Noname manuscript No.
(will be inserted by the editor)

Towards Effective and Efficient Temporal Activity Detection

Da Zhang · Xiyang Dai · Xin Wang · Yuan-Fang Wang

Received: date / Accepted: date

Abstract Detecting temporal activities in untrimmed video streams is an important and challenging task. To solve this problem, we need to not only recognize the activity categories, but also localize the temporal span, i.e., the start and end time of each activity instance. In this paper, we present a novel Single-Shot Multi-Span Detector for effectively and efficiently detecting temporal activities in long, untrimmed videos using a simple end-to-end, fully three-dimensional convolutional (Conv3D) network. Our approach, named S^{3}D, encodes the entire video stream and discretizes the output space of temporal activity spans into a set of default spans over different temporal locations and scales. At the prediction time, S^{3}D predicts scores for the presence of activity categories in each default span as well as temporal adjustments relative to the span location to produce the precise activity duration. Additionally, S^{3}D combines predictions from multiple feature maps with different temporal resolutions to naturally handle activities of different lengths, and temporal contextual information is further exploited by appropriately fusing multi-scale feature maps. Unlike state-of-the-art systems that require a separate proposal and classification stage, S^{3}D encapsulates all computation in a single-shot, end-to-end detection framework, making it simpler, faster and more robust. We evaluate the proposed approach on two challenging public benchmarks THUMOS'14 and ActivityNet, where S^{3}D achieves state-of-the-art performance and efficiency in both.

Keywords Temporal Activity Detection · Single-shot Detector · 3D Convolutional Network

D. Zhang
University of California at Santa Barabra, CA, 93106
Tel.: +1(805)689-6971
E-mail: dazhang@cs.ucsb.edu

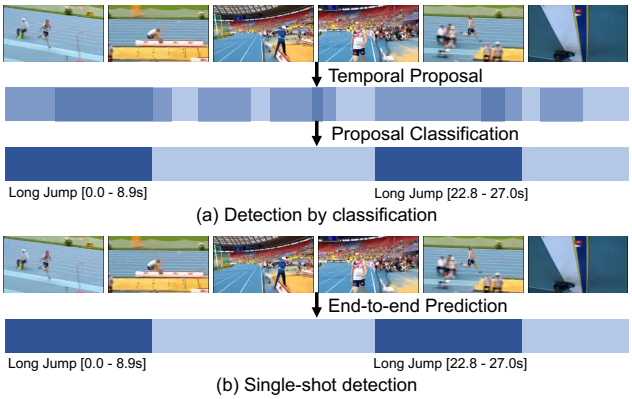

Fig. 1 We consider the problem of temporal activity detection in long untrimmed video streams. (a) Conventional two-stage detector where temporal proposal generation and proposal classification are designed as separate components; (b) The proposed single-shot end-to-end temporal detection framework to directly produce prediction results.

1 Introduction

Temporal activity detection has drawn increasing interests in both academic and industry communities due to its vast potential applications in security surveillance, video analytics, videography, etc. Different from activity recognition, which only aims at classifying the categories of manually trimmed video clips, activity detection is for localizing and recognizing activity instances from long, untrimmed video streams. It is substantially more challenging, as it is expected to handle activities with variable lengths, predicting not only the activity category but also the precise temporal boundary of each instance. Advances in deep Convolutional Neural Network (CNN) have led to significant progress in video analysis over the past few years. While the performance of activity recognition has improved a lot [36,37,30,9,

40,34], the detection performance still remains unsatisfactory [38,43,29].

A typical framework used by many state-of-the-art systems [24,29,38,28] is *detection by classification* (Fig. 1), where temporal proposals are generated by sliding windows [24,29] or advanced proposal methods [39,4] and separate activity classifier is then applied to predict the final detection results. In order to predict the precise starting and ending times, a recent work [28] applies deconvolutional networks to generate per-frame prediction scores for each temporal proposal and adjusts temporal boundaries accordingly. However, there are certain limitations to these frameworks: (1) Temporal proposal generation and classification are independent processes and optimized separately using different networks, resulting in sub-optimal performance, (2) the classification network only takes the proposal frames as input, thus forbidding it to see a larger temporal context which can be beneficial, and (3) a two-stage approach is usually slow due to inefficient proposal methods and duplicate operations repeated in the proposal and classification stages.

In this paper, We propose a *Single-shot multi-Span Detector (S^3D)*, a simple yet novel fully Conv3D network for effective and efficient temporal activity detection in continuous untrimmed video streams. As illustrated in Fig. 2, S^3D produces a fixed-size collection of temporal spans and scores for the presence of activity class instances in those spans, followed by a temporal non-maximum suppression (NMS) step to generate the final detection results. S^3D is a highly-unified network by eliminating explicit temporal proposal and classification stages and solving the detection problem in one single shot. We set multi-scale default spans at feature maps with different temporal resolutions to naturally handle activities of different lengths, and the feature maps are further augmented with temporal contextual information by appropriately fusing multi-scale feature representations. Furthermore, we predict the temporal offsets to adjust each default span in order to predict precise temporal boundaries.

Our framework is based on the successful Conv3D filters [34,5]. While the original Conv3D filter is applied upon a short video segment for activity recognition, the convolutional nature allows it to be applied on video with an arbitrary length. In S^3D, the network takes as input the whole video stream and applies a sequence of Conv3D filters to extract features and produce prediction results, allowing our scheme to see a larger temporal context and produce better detection results. The whole network is end-to-end trainable with a joint loss to directly maximize the detection performance.

Comparing to existing state-of-the-art systems, our S^3D is much simpler, faster and more robust. In summary, our work makes the following contributions:

- We introduce S^3D, a single-shot end-to-end activity detection model based completely on Conv3D networks that can effectively predict both the precise temporal boundaries and confidence scores of multiple activity categories in untrimmed videos.
- We demonstrate experimentally that our S^3D achieves state-of-the-art performance on temporal activity detection task on both THUMOS'14 and ActivityNet benchmarks.
- To the best of our knowledge, by eliminating explicit proposal generation step and formulating the temporal activity detection in a single-shot solution with fully Conv3D layers, our detector achieves the fastest run-time speed on a single GTX 1080 Ti GPU with 1271 frame per seconds (FPS).

This paper significantly extends our recent BMVC 2018 conference paper in [45], and the improvements are multiple folds with in-depth discussion. First, we provide a deeper insight into the proposed single shot multi-span detector in a more general setting. In this view, our framework is compatible with a wide range of Conv3D-based networks. Second, we offer detailed in-depth discussions in all sections in this paper including but not limited to related works, network optimization strategies, prediction methods, quantitative and qualitative experiment results. Third, based on the original architecture proposed in [45], we further exploit temporal contextual information by explicitly fusing multi-scale temporal features and obtains the new state-of-the-art for the temporal detection task. Last but not least, extensive experiments are conducted for thoroughly and insightfully examining the effectiveness of single-shot multi-span detector. The results on an additional large-scale ActivityNet [8] dataset show the generalization of our method.

The rest of the paper is organized as follows. In Section 2, we first present representative works related to our approach including *activity recognition*, *object detection* and *temporal activity detection*. Then we show details of our approach in Section 3. In Section 4, we provide the training details of our network. In Section 5, we offer the network prediction and post-processing details. In Section 6, both quantitative and qualitative experimental results and analysis of the proposed approach are provided. Experiments on two challenging public temporal activity detection benchmarks, namely THUMOS'14 [17] and ActivityNet [8], with comparison to the state-of-the-art methods, distinctly demonstrate the efficiency, robustness, and effectiveness of our

single-shot multi-span detector. The experimental results show that our method achieves the new state-of-the-art performance and efficiency. Finally, the concluding remarks are summarized in Section 7.

2 Related Work

In this section, we review relevant works in activity recognition, object detection, and temporal activity detection. Other works on spatial-temporal activity detection and video segmentation are beyond the scope of this paper.

Activity Recognition. Activity recognition is an important research topic for video analysis and has been extensively studied in the past few years. Earlier methods were often based on hand-crafted visual features such as improved Dense Trajectory (iDT) [36,37] consisting of HOG, HOF, MBH features extracted along dense trajectories, feature encoding with Fisher Vector (FV) [25,23], VLAD [16], etc. In the past few years, tremendous progress has been made due to the introduction of large datasets [8,17] and the developments of deep neural networks [34,9,40,35,30,5,26]. Two-stream network [30] learned both spatial and temporal features by operating the network on single frames and stacked optical flows using 2D CNN such as AlexNet [19], VGG [31] and ResNet [14]. 3D CNN architecture called C3D [34] used Conv3D filters to capture both spatial and temporal information directly from raw video frames. More recently, improvements on top of the C3D architecture [35,5,26] as well as advanced temporal building blocks such as non-local modules [41] were proposed to further boost the performance. However, the assumption of well-trimmed videos where the activity of interest lasts for the entire video duration limits the application of these approaches in real scenarios, where the videos are usually long and untrimmed. Although they do not consider the difficult task of localizing activity instances, these methods are widely used as the backbone network for the detection task.

Object Detection. Activity detection in untrimmed videos is closely related to object detection [13,27,21] in spatial images, where detection is performed by classifying region proposals into foreground classes or a background class. Earlier work [13] relied on an external region proposal method and trained a CNN classifier to classify each proposed region. Faster-RCNN [27] incorporated a region proposal network and RoI pooling to jointly generate and classify region proposals with a single network, resulting in a large improvement of the accuracy and efficiency. SSD [21] completely eliminated proposal generation and subsequent feature re-sampling stages and encapsulated all computation in a single network to directly output object locations and confidence scores. Our network is inspired by SSD [21] and adopt similar design philosophies for temporal activity detection. Like SSD [21], our S^3D model is also designed for both accuracy and efficiency in a single-shot operation.

Temporal Activity Detection. Unlike activity recognition, the detection task focuses on learning how to detect activity instances in untrimmed videos with annotated temporal boundaries and instance category. The problem has recently received significant research attention due to its potential application in video data analysis.

Early approaches on activity detection mainly used temporal sliding windows as candidates and classified them with activity classifiers trained on multiple features [23,10,15,22,33]. They typically extract iDT features or pre-trained DNN features, and globally pool these features within each window to obtain the input for the SVM classifiers. However, these approaches might be computationally inefficient, because one needs to apply each activity classifier exhaustively on windows of different sizes at different temporal locations throughout the entire video.

Inspired by the success of region-based detectors in object detection [13], many recent works adopt a two-stage, proposal-plus-classification framework [3,28,29,46,11], i.e. first generating a sparse set of class-agnostic segment proposals from the input video, followed by classifying the activity categories for each proposal. A large number of these works focus on designing better proposal schemes [3,46,11], while others focus on building more accurate activity classifiers [28,29,46]. However, most of these methods do not afford end-to-end training on either the proposal or classification stage. And the proposals are typically selected from sliding windows of predefined scales, where the boundaries are fixed and may result in imprecise localization results.

Along this line of attack, Faster-RCNN is the latest region-based object detector which is composed of end-to-end trainable proposal and classification networks, and applies region boundary regression in both stages. A few very recent works have started to apply such architecture to temporal activity detection [42,7,6], and demonstrated competitive detection accuracy. R-C3D [42] is a classic example that closely follows the original Faster-RCNN in many design details. Dai *et al.* [7] explicitly modeled temporal contextual information into the proposal stage. Chao *et al.* [6] proposed to use a multi-tower network with temporal contexts to further improve the detection performance. However, all these methods require a separate temporal proposal and activity classification method.

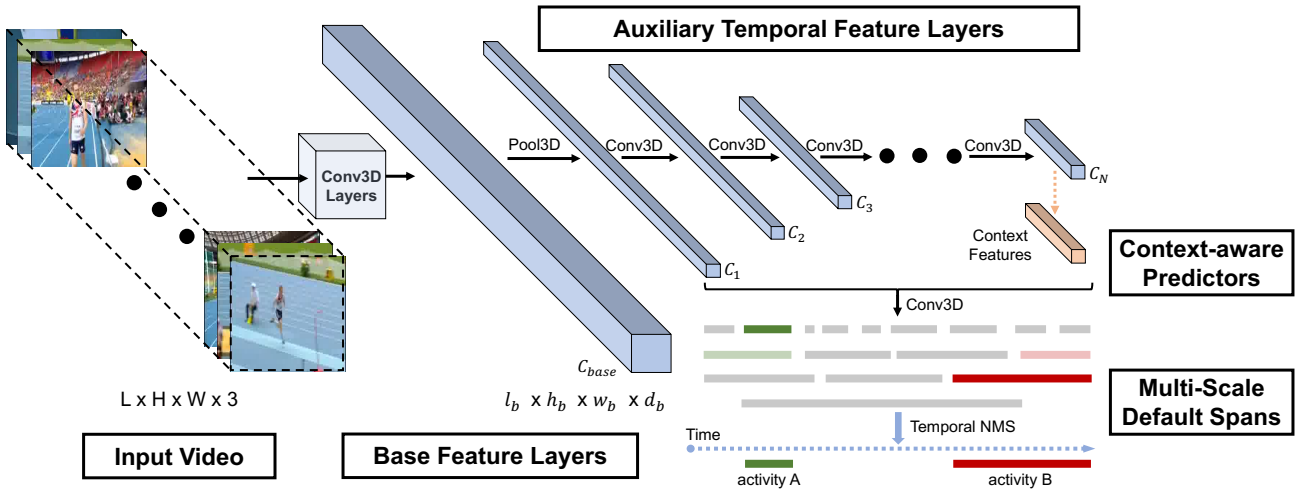

Fig. 2 S^{3}D network architecture: Our network takes an untrimmed video $\nu = \{I_i\}_{i=1}^{L}, I_i \in \mathbb{R}^{H \times W \times 3}$ as input and computes base features using a standard Conv3D-based network truncated before the first fully-connected layer. We add auxiliary Conv3D layers on top of C_{base} to produce a temporal feature hierarchy with multi-scale default spans at each layer. For each temporal feature map cell, we predict class confidence scores and location offsets with a set of Conv3D filters. Temporal NMS is applied to produce the final detection results. Please refer to Fig. 3 and Fig. 4 for a detailed illustration of multi-scale default spans and temporal contextual modeling.

Most recently, several attempts were made towards single-shot temporal activity detection: SSAD [20] proposed to directly predict activity instances in untrimmed videos with a separate feature extraction and detection network. SS-TAD [2] have investigated the use of gated recurrent memory module in a single-stream detection framework. Our approach is one of the first within this group to propose a highly-integrated detection architecture. With a simple end-to-end Conv3D network which learns directly from raw video frames to produce final detection outputs, S^{3}D is able to jointly optimize feature representation and prediction layers, resulting in a simple, fast and robust architecture achieving both state-of-the-art performance and fast run-time speed.

3 Approach

We introduce a *Single-shot multi-Span Detector (S^{3}D)*, a simple yet novel fully Conv3D network for activity detection in continuous untrimmed video streams. The S^{3}D approach, illustrated in Fig. 2, is based on a feed-forward fully Conv3D network that produces a fixed-size collection of temporal spans and scores for the presence of activity class instances in those spans, followed by a temporal NMS step to generate the final detection results.

Our model consists of four major components: base feature layers, auxiliary temporal feature layers, multi-scale default spans and context-aware convolutional predictors. The early network layers are based on a standard architecture used for activity classification (truncated before the first fully-connected layer) to extract high-level features given an input video stream, which we call the *base feature layers* (Section 3.1). We then add *auxiliary temporal feature layers* (Section 3.2) to the end of the based feature layer to generate rich spatial-temporal feature hierarchies. These layers decrease in temporal dimension progressively and allow predictions of temporal spans at different locations and scales. We associate a set of default temporal spans with each feature map cell and the default spans tile the tile the feature map in a convolutional manner, which we denote as the *multi-scale default spans* (Section 3.3). At each feature map cell, we predict the offsets relative to the default span in the cell, as well as the confidence scores that indicate the presence of an activity instance in each of those spans. These are done by adding *context-aware convolutional predictors* (Section 3.4) on top of each cell. We now describe each part of S^{3}D in detail.

3.1 Base Feature Layer

The base feature layer is used to extract compact yet representative spatial-temporal features from a given input video stream. Like two-dimensional convolution (Conv2D) filters have been widely used to extract features for images, we use Conv3D filters which convolves in both spatial and temporal dimensions to generate rich feature hierarchies for videos.

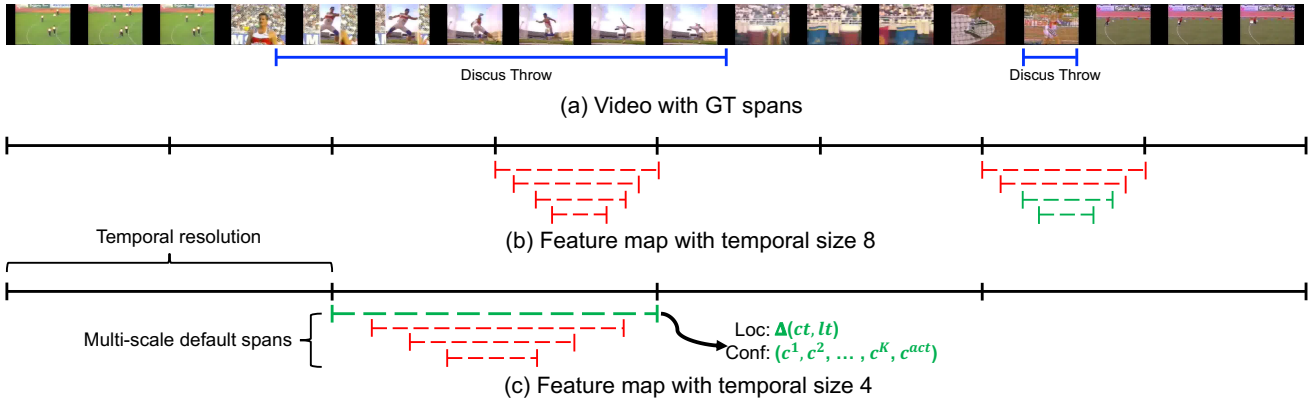

Fig. 3 S^{3}D framework. (a) Input video with temporal ground-truth annotations. We evaluate a small set (*e.g.* $R = 4$) of multi-scale default spans at each location in several feature maps with different temporal resolutions. For each default span, we predict both the temporal offsets and the confidences for presence of activity and all activity categories. At training time, we match the default spans to the ground truth spans.

In more detail, the input of our network is a long, untrimmed video with an arbitrary length. We denote a video ν as a series of RGB frames $\nu = \{I_i\}_{i=1}^{L}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$ is the i-th input frame, L is the total number of frames, H and W are the height and width of each frame, respectively. We apply the Conv3D-based architectures [34,5,35] for activity recognition as our base feature layer as it has been proven as an effective building block in prior works [42,6]. Specifically, we adopt the Conv3D layers (truncated before the first fully-connected layer) of a classification network [34,5, 35] to generate a feature map $C_{base} \in \mathbb{R}^{l_b\times h_b\times w_b\times d_b}$ where d_b is the output feature dimension, l_b, h_b and w_b are derived from the equivalent convolution strides given a specific classification network. we use C_{base} as our base feature since it is a rich yet compact spatial-temporal representation of the input video stream. More importantly, as the backbone network of an object detector can be easily altered with different Conv2D networks, our base feature layer is also compatible with a wide range of video encoding architectures.

3.2 Auxiliary Temporal Feature Layers

To allow the model to predict variable scale temporal spans, we add temporal feature layers to the end of the base feature layer. Specifically, we first down sample C_{base} both spatially and temporally to reduce the computational overhead while keep sufficient spatial-temporal information. We then add auxiliary Conv3D layers to produce a sequence of feature maps $\{C_i\}_{i=1}^{N}$, $C_i \in \mathbb{R}^{l_i\times h\times w\times d}$ that progressively decrease in temporal dimension while keeping the same spatial resolution, where N is the total number of temporal feature maps each with a temporal dimension l_i and spatial dimension $h\times w$. For a simple and efficient design, we set each feature map to have the same feature dimension d and the temporal dimension in between follows $l_i = 2l_{i+1}$.

The detailed network configurations are illustrated in Fig. 2, we stack Conv3D layers with temporal kernel size 3 to extend the temporal receptive field and the stride is set to 2 for progressively decreasing the temporal dimension. We also add bottleneck Conv3D layers to help prevent over-fitting and improve run-time efficiency. Simplicity is central to our design and we have found that our model is robust to many design choices. We have experimented with more enhanced building blocks such as dilated convolution [44] and observed marginally better results. Designing better network blocks is not the focus of this paper, so we opt for the simple design described above.

The network is intrinsically simple by only applying Conv3D filters, but builds a rich feature hierarchy by summarizing a continuous video stream in multiple temporal resolutions, allowing us to add default temporal spans at certain layers to obtain temporal predictions at multiple scales.

3.3 Multi-scale Default Spans

To handle different activity locations and scales, [29] suggests processing the video at different segment levels and combining the results afterward, while [2] uses a gated recurrent network to assign a number of anchors at different time steps. However, by utilizing feature maps from several different layers in a single network for prediction we can mimic the same effect, while also sharing parameters across all temporal scales. We use

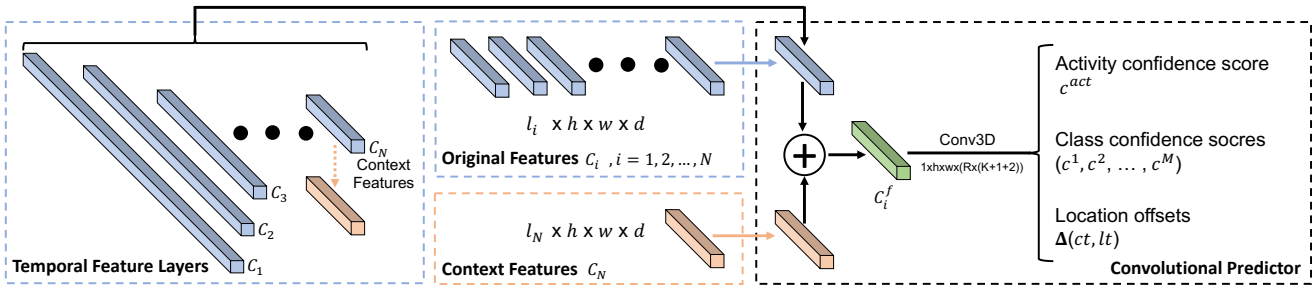

Fig. 4 Illustration of the context-aware convolutional predictor. Every temporal feature map cell at all layers is enhanced by the last feature map C_N which summarizes the whole video content. For each temporal feature map cell with contextual information, we predict 1 activity confidence score, K class confidence scores and 2 location offsets with a convolutional predictor. The convolutional predictor is a single Conv3D layer with filter size $1 \times h \times w$ to produce a prediction vector of size $R \times (K + 1 + 2)$.

feature maps with different temporal resolutions for detection since earlier feature maps have higher resolution and capture finer details of the input video, and deeper feature maps have larger receptive fields and contain more temporal contexts. So we let the earlier feature maps detect short activity spans and the deeper feature maps detect long activity instances.

In our design, we use $\{C_i\}_{i=1}^N$ as our temporal feature maps and associate a set of multi-scale default spans with each temporal feature map cell. We design the tiling of default spans so that specific feature maps learn to be responsive to particular locations and lengths of the activities. Regrading a temporal feature map C_i with temporal length l_i, the scale of the default spans for this feature map is set as $S_i = \frac{1}{l_i}$ (as the input video length is normalized to 1). We impose different scale ratios for the default spans, and denote them as $r \in \{0.25, 0.5, 0.75, 1.0\}$. We can compute the length ($l_i^r = S_i \cdot r$) for each default span, and we set the center of each default span to $\frac{j-0.5}{l_i}$, where j indicates the j-th temporal feature cell, $j \in [1, l_i]$. So for an temporal feature map with length l_i and R different scale ratios ($R = 4$), the number of default spans is $l_i \cdot R$. A concrete example is illustrated in Fig. 3 where two feature maps with temporal dimension 8 and 4 are shown along with the input video. Note that R and N can be chosen arbitrarily to accommodate for a variety of practical scenarios.

By combining predictions for all default spans with different scales from all locations of multi-scale feature maps, we have a diverse set of predictions, covering various activity locations and lengths. For example, in Fig. 3, the longer Discus Throw instance is matched to a default span in (c), but not to any default spans in (b). This is because those spans have different center locations and scales, and do not match the longer activity instance, and therefore are considered as negatives during training.

3.4 Context-aware Convolutional Predictors

Temporal contextual information has been shown to be critical for temporal activity detection [7,6] since it provides strong semantic cues for identifying the activity class. For example, seeing a video during summer Olympics is a strong indicator for sports activities while it is more likely to detect daily activities in a household video. As a result, it is critical to encode the temporal context features in the activity detection pipeline. In addition to our conference paper [45] where features are directly used to produce the final results, we explicitly exploit context features to model more complex temporal dynamics. Below, we provide detail of our approach.

Our multi-scale feature hierarchy can easily incorporate contextual information since it naturally summarizes temporal information at different scales. In Section 3.3, we showed different layers with different temporal resolutions can be responsive to varying temporal locations and scales. However, this only extracts the features within each default span, and overlooks the temporal contexts. To ensure the context features are used for span classification and boundary regression, we combine the feature map at each layer C_i with the last feature map C_N which summarizes the whole video content. Technically, we tile C_N to have the same temporal dimension with C_i and construct $C_i^f = C_i + C_N$. We exploit temporal contexts at all layers in our network, thus, each temporal feature cell is enhanced by the global temporal information. We illustrate this mechanism in Fig. 4.

Each temporal feature layer can produce a fixed set of detection predictions using a set of Conv3D filters. These are indicated on top of the feature network architecture in Fig. 4. For a temporal feature map $C_i^f \in \mathbb{R}^{l_i \times h \times w \times d}$, the basic operation for predicting parameters of a potential temporal detection is a $1 \times d \times w$

kernel that produces scores for activity presence and categories, or temporal offsets relative to the default location and scale. At each of the l_i temporal locations where the kernel is applied, it produces an output value. The temporal offset values are measured relative to a default span location in each temporal feature map cell. Specifically, for each default span at a given temporal location, we compute K positive class confidence scores plus one activity confidence score and two temporal offsets. This results in a total of $(K+1+2) \times R$ filters that are applied around each location in the feature map, yielding $(K + 1 + 2) \times R \times l_i$ outputs for a temporal feature map C_i^f. Each default span gets a prediction score vector $v_{pred} = (c^1, c^2, ..., c^K, c^{act}, \Delta ct, \Delta lt)$ with length $K + 1 + 2$, where c^{act} is a class-agnostic confidence score to estimate the presence of activity, c^1 to c^K are used to predict default span's category and $\Delta ct, \Delta lt$ are temporal offsets.

4 Optimization

The key step of training S^{3}D is that the ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs. Once this assignment is determined, the loss function and back propagation are applied. Training also involves training data construction and augmentation as well as hard negative mining strategies.

4.1 Training Data Construction and Augmentation

In theory, because S^{3}D is a fully Conv3D network, it can be applied to an input of arbitrary size. Therefore, our S^{3}D network can operate on videos of variable lengths. In practice, due to GPU memory limitations, we slide a temporal window of size L frames on the video and feed each windowed segment individually into the S^{3}D network to obtain temporal detections. Although the input window size is fixed, we decode the input video stream with a small FPS, allowing the network to encode enough temporal contexts for precisely detecting activity instances. During training phase, we only keep windows that have at least one ground truth activity instance. Therefore, given a set of training videos, we obtain a training collection of windows with temporal activity annotations inside each windowed video segment.

To make the model more robust to various activity locations and scales, we further improve the training dynamics by augmenting the training videos via following strategies:

- We slide the window on the input video stream with a small step size. Thus, given a certain video stream, the same activity instance can appear in different locations relative to the start and end time of certain windows.
- Similar to [34], we resize each frame to have a spatial size of 128×171 and randomly crop each window into $L \times 112 \times 112$ for spatial and temporal jittering.
- We pad extra zero frames (black image with all RGB values set to 0) to the end of certain window if the video length is less than our window size.

4.2 Matching Strategy

During training, we need to determine which default spans correspond to a ground truth detection and train the network accordingly. Specifically, for each default span, we compute the Intersection-over-Union (IoU) score with all ground truth instances. If the highest IoU score is higher than 0.5, we match the default span with the corresponding ground truth span and regard it as positive, otherwise negative. So a ground truth instance can match multiple default spans while a default span can only match one ground truth instance at most. This simplifies the learning problem, allowing the network to predict high scores for multiple overlapping default spans.

4.3 Hard Negative Mining

After the matching step, most of the default spans are negatives. This introduces a significant imbalance between the positive and negative training examples. Instead of using all the negative examples, we sort them using the highest activity confidence loss for each default span and pick the top ones so that the ratio between the negatives and positives is nearly $1:1$. We found that this leads to faster optimization and a more stable training.

4.4 Training Objective

The training objective of S^{3}D is to solve a multi-task optimization problem. Let $x_{ij}^k = \{1, 0\}$ be an indicator for matching the i-th default span to the j-th ground truth span of category $k \in [1, K]$, and s_i be the highest IoU score with any ground truth spans. The overall objective loss function is a weighted sum of the localization loss (loc), class confidence loss (conf) and activity confidence loss (act):

$$\mathcal{L} = \mathcal{L}_{loc}(x, t, g) + \alpha \mathcal{L}_{conf}(x, c) + \beta \mathcal{L}_{act}(s, c) \quad (1)$$

where α and β are the weight terms used for balancing each part of the loss function.

The localization loss is a Smooth L1 loss [12] between the predicted temporal offsets (t) and the ground truth span parameters (g). In temporal domain, we regress to offsets for the center (ct) of the default span (d) and for its length (lt):

$$\mathcal{L}_{loc}(x,t,g) = \frac{1}{N_{pos}} \sum_{i}^{N_{pos}} \sum_{m \in \{ct,lt\}} x_{ij}^{k} smooth_{L1}(t_i^m - \hat{g}_j^m) \tag{2}$$

where N_{pos} is the number of positive matching default spans in a batch, and the temporal offset parameters $\hat{g}_j^m$ are defined similarly like the bounding box offset in object detection [12]:

$$\hat{g}_j^{ct} = \Delta ct_i = (g_j^{ct} - d_i^{ct})/d_i^{lt} \qquad \hat{g}_j^{lt} = \Delta lt_i = \log(\frac{g_j^{lt}}{d_i^{lt}}) \tag{3}$$

where g_j^{ct}, d_i^{ct} are the centers and g_j^{lt}, d_i^{lt} are the lengths for the ground truth span and the matching default temporal span respectively.

The class confidence loss is a softmax loss over multiple class confidences (c):

$$\mathcal{L}_{conf}(x,c) = -\frac{1}{N_{pos}} \sum_{i}^{N_{pos}} x_{ij}^{k} \log(\hat{c}_i^k) \tag{4}$$

where $\hat{c}_i^k = \frac{\exp(c_i^k)}{\sum_k \exp(c_i^k)}$ is the softmax probability for the ground truth class of this instance. The class confidence loss is only used to distinguish between multiple positive classes not including the background. We use another activity confidence score to predict activity class agnostic scores.

The activity confidence loss is a binary classification loss using sigmoid cross-entropy. Rather than using a hard ground truth score for positive (1) and negative (0), we use the IoU score s_i as ground truth for each default span. This helps the training procedure since positive default spans are assigned different confidence levels based on its overlap with the ground truth span. We define the activity confidence loss as:

$$\mathcal{L}_{act}(s,c) = -\frac{1}{N} \sum_{i}^{N} (s_i \log(c_i^{act}) + (1 - s_i) \log(1 - c_i^{act})) \tag{5}$$

where N is the number of total training default spans in a batch and $N = N_{pos} + N_{neg}$; c_i^{act} is the predicted activity confidence score. Note that we separate the activity confidence score and class confidence scores via two separate losses. Comparing to only having one softmax classification loss containing all positive classes and one background class, we find this configuration is more robust, leads to better validation performance and makes the network architecture more flexible.

5 Prediction

Activity prediction in S^{3}D is a single shot with one forward pass of the network. Given an input video stream, we predict activity confidence score, class confidence scores and temporal location offsets at each default span. The temporal location offset is in the form of relative displacement as described in Equation 3, which is applied on the default span to predict accurate start time and end time. Then the spans with low activity confidence scores will be filtered out and the remaining spans are refined via temporal NMS operation. The final predictions are assigned the activity label with the highest class confidence score.

In more detail, for each default span, we predict K class confidence scores, 1 activity confidence score and 2 temporal location offsets. We denote the predicted vector as $v_{pred}^i = \{c_i^1, ..., c_i^K, c_i^{act}, \Delta ct_i, \Delta lt_i\}$, where i indicates the i-th default span. The post processing steps are as follows:

1. First, we adjust the temporal boundaries of each default span by applying Δct_i and Δlt_i and generate the adjusted start time t_i^s and end time t_i^e.
2. Then, we remove those default spans whose activity confidence score c_i^{act} is less than a threshold, and keep the remaining ones. We set the threshold to 0.1 from cross-validation.
3. Given $\{t_i^s, t_i^e, c_i^{act}\}$ of each remaining span, we apply temporal NMS with threshold value 0.5 to further filter out low confidence spans.
4. Finally, to generate the predicted results, we choose the largest entry from c_i^1 to c_i^K denoted as c_i^{class} and assign the activity label $class$ as the predicted label for this default span. The final confidence score is computed as $c_i^{final} = c_i^{class} \times c_i^{act}$. Thus, each final detection span is denoted as $\{class, c_i^{final}, t_i^s, t_i^e\}$.

6 Experiments

We evaluate the proposed approach on two recent large-scale datasets for the temporal activity detection task: THUMOS'14 [17] and ActivityNet [8]. In this section we first introduce these datasets and our implementation details and then compare the performance of S^{3}D with other state-of-the-art approaches. Finally, we investigate the impact of different components via a set of ablation studies and provide qualitative examples.

6.1 Datasets

THUMOS'14 [17]. The temporal activity detection task of THUMOS'14 dataset is challenging and widely used. The official training set is the UCF101 [32] dataset including trimmed videos of 101 categories, the validation and test set contains 1010 and 1574 untrimmed videos, respectively. For temporal activity detection, over 20 hours of video and 20 activity categories are involved and annotated temporally, resulting in 200 validation and 213 test untrimmed videos. Following the standard practice, we train our models on the validation set and evaluate them on the testing set.

Evaluation Metrics. We follow the conventional metrics used in THUMOS'14, comparing the Average Precision (AP) for each activity category and calculating mean average precision (mAP) for evaluation. A prediction instance is correct if it gets the same category with the ground truth instance and its temporal IoU is larger than the IoU threshold. On THUMOS'14, the IoU thresholds are $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. The mAP at 0.5 is used for comparing results from different methods.

ActivityNet [8]. The large-scale ActivityNet is a recently released dataset which contains 200 different types of activities and a total of 849 hours of videos collected from YouTube. ActivityNet is the largest benchmark for temporal activity detection to date in terms of both the number of activity categories and number of videos, making the task particularly challenging. There are two versions, and we use the latest version 1.3 which contains $19,994$ untrimmed videos in total and is divided into three disjoint subsets, training, validation and testing by a ratio of $2:1:1$. On average, each activity category has 137 untrimmed videos. Each video on average has 1.41 activities which are annotated with temporal boundaries. Since the ground truth annotations of test videos are not public, following traditional evaluation practices on this dataset, we report performance on the validation subset.

Evaluation Metrics. ActivityNet dataset has its own convention of reporting performance metrics. We follow their conventions, reporting mAP at different IoU thresholds 0.5, 0.75 and 0.95. The average of mAP values with IoU thresholds $[0.5 : 0.05 : 0.95]$ is used to compare the performance between different methods.

In our experiment, we compare our approach with the state-of-the-art methods on both THUMOS'14 and ActivityNet, and perform ablation studies on THUMOS'14.

6.2 Implementation Details

THUMOS'14. S^{3}D takes as input $L = 256$ raw video frames, since this allows us to train the network on a single GPU with efficient mini-batch training. We decode each video at 8 FPS and follow the training data construction and augmentation strategy discussed in Section 4.1 to produce a collection of training windows. Thus, each window contains 32 seconds of a video stream and this is motivated by the fact that more than 99% of activity instances in THUMOS'14 are less than 32 seconds. We use the last Conv3D feature map in C3D [34] as the base feature map and apply 3D max pooling to reduce the size to $32 \times 4 \times 4 \times 512$, we append $N = 6$ more temporal feature layers $\{C_i\}_{i=1}^{6}$ with temporal dimension $\{32, 16, 8, 4, 2, 1\}$ and associate a set of default spans at each temporal feature cell with $R = 4$ ratios $\{0.25, 0.5, 0.75, 1.0\}$, resulting in 252 default spans in total. Besides, each temporal feature map is also augmented with context features as described in Section 3.4 to construct $\{C_i^f\}_{i=1}^{6}$; the default spans correspond to spans of duration between 0.25s and 32s uniformly distributed at different temporal locations. We also add l2 normalization layers along the feature dimension on top of each temporal feature map before the context-aware convolutional predictors, and the default scale is set to 10.0 and can be jointly learned with the network. We initialize base feature layers with C3D weights pre-trained on Sports-1M by the authors in [34], and other layers from scratch.

ActivityNet. In addition to our conference paper [45], we also apply our S^{3}D to the recent ActivityNet dataset. As the length of activity instances in ActivityNet is much longer than THUMOS'14, instead of decoding the videos at a fixed FPS, we uniformly sample $L = 256$ raw video frames from the input video. Thus, the longest default span corresponds to the whole video duration. We apply a Residual Conv3D model [35] as the backbone network instead of using the C3D network. Since this is motivated by the fact that ActivityNet contains a lot more activity classes compared to THUMOS'14 thus stronger base features are needed. Following similar designs, we also add residual links in our auxiliary temporal feature layers. Other implementation details stay the same with those of THUMOS'14. We initialize base feature layers with Res3D weights pre-trained on Kinetics [5] and other layers from scratch.

Our implementation is based on TensorFlow [1] and we use Adam [18] to learn the network parameters with the end-to-end loss function as described in Section 4.4. As a speed-accuracy trade-off on both datasets, we freeze the early network layers and allow all the other layers of S^{3}D to be trained with a fixed learning rate of 0.0001.

Table 1 Temporal activity detection mAP on THUMOS'14. The top performing methods in existing papers are shown. S^{3}D achieves state-of-the-art performance at different overlap threshold. (- indicates that results are unavailable in the corresponding papers).

IoU threshold	0.3	0.4	0.5	0.6	0.7
S-CNN [29]	36.3	28.7	19.0	10.3	5.3
CDC [28]	40.1	29.4	23.3	13.1	7.9
SSAD [20]	43.0	35.0	24.6	-	-
TCN [7]	-	33.3	25.6	15.9	9.0
R-C3D [42]	44.8	35.6	28.9	-	-
SS-TAD [2]	40.1	-	29.2	-	9.6
S^{3}D	47.9	41.2	32.6	23.3	14.3
S^{3}D-context	**48.4**	**42.4**	**34.3**	**25.1**	**15.0**

On THUMOS'14, we use a mini-batch size of 20 and the network is trained for 20 epochs. On ActivityNet, we use a mini-batch size of 10 and the network is trained for 30 epochs.

6.3 Comparison with State-of-the-art

We compare our S^{3}D with other state-of-the-art methods on THUMOS'14 [17] and ActivityNet [8], and report the performances using the metrics described above. Note that the average activity duration in THUMOS'14 and ActivityNet are 4 and 50 seconds. And the average video duration are 233 and 114 seconds, respectively. This reflects the distinct natures of these datasets. Hence, strong adaptivity is required to perform consistently well on both datasets.

THUMOS'14. The comparison results between our S^{3}D and other top-performing methods are summarized in Table 1 with multiple IoU thresholds varying from 0.3 to 0.7, and our original S^{3}D without explicit temporal contextual modeling already significantly outperforms all previous state-of-the-art methods. Furthermore, S^{3}D improves the state-of-the-art by a large margin when the IoU thresholds are set at higher levels (0.5 to 0.7), indicating its superior ability to predict precise temporal boundaries of different activities. Moreover, adding temporal context features further boosts the performance at all evaluation thresholds, setting a new state-of-the-art compared to the original results reported in our conference paper [45]. Note that, as described in Section 3.4, temporal contextual modeling is efficently implemented by fusing features at multiple levels without increasing model complexity or inducing further computational overhead.

We also compare our S^{3}D with two other representative methods: S-CNN [29] and R-C3D [42], measuring the AP for each class in THUMOS'14 at IoU threshold 0.5. The results are shown in Table 2, and note that we compare with the original S^{3}D model as contextual features are not integrated in S-CNN and R-C3D. Our S^{3}D outperforms the existing methods in 11 out of 20 classes on THUMOS'14 and improves the AP by a large margin in most cases (*e.g. Pole Vault*, *Long Jump*, *Javelin Throw*, *High Jump*, *Golf Swing*, *Diving*, *Cliff Diving*). For the other 9 activities, S^{3}D performs reasonably well and achieves similar AP compared to the existing methods (*e.g. Volleyball Spiking*, *Shotput*, *Baseball Pitch*).

In comparison with the proposed S^{3}D model: previous systems on top of C3D networks (S-CNN [29], CDC [28]) largely relies on good temporal proposals generated by external proposal methods and only processes a small number of frames at a time, restricting them from directly optimizing the detection performance. R-C3D [42] is able to process a long video stream and predict multi-scale activity instances, but it only applies anchors on a single feature map with fixed temporal dimension. With the proposed S^{3}D framework, we jointly optimize the feature representation and

Table 2 Per-class AP at IoU threshold 0.5 on THUMOS'14.

Activity	S-CNN [29]	R-C3D [42]	S^{3}D
Volleyball Spiking	4.6	**5.6**	4.1
Throw Discus	24.4	29.2	**29.5**
Tennis Swing	**19.3**	16.6	9.9
Soccer Penalty	19.2	15.8	**19.3**
Shotput	12.1	**19.4**	17.1
Pole Vault	32.1	42.7	**60.1**
Long Jump	34.8	57.4	**78.5**
Javelin Throw	18.2	47.0	**60.7**
High Jump	20.0	30.9	**47.2**
Hammer Throw	19.1	43.2	**50.8**
Golf Swing	18.2	16.1	**30.3**
Frisbee Catch	15.3	**20.1**	12.5
Diving	17.6	26.2	**57.9**
Cricket Shot	**13.8**	10.9	4.0
Cricket Bowling	15.7	**30.6**	10.8
Cliff Diving	27.5	49.2	**62.5**
Clean and Jerk	24.8	27.9	**38.0**
Billiards	7.6	**8.3**	6.1
Basketball Dunk	20.1	**54.0**	34.5
Baseball Pitch	14.9	**26.1**	17.6
mAP@0.5	19.0	28.9	**32.6**

Table 3 Activity detection results on ActivityNet v1.3 validation subset. The performances are measured by mean average precision (mAP) for different IoU thresholds and the average mAP of IoU thresholds from 0.5 : 0.05 : 0.95 (- indicates that results are unavailable in the corresponding papers).

IoU threshold	0.5	0.75	0.95	Average
R-C3D [42]	26.80	-	-	12.70
TCN [7]	36.44	21.15	3.90	-
Chao *et al.* [6]	**38.23**	18.30	1.30	20.22
S^{3}D	34.51	21.41	2.55	21.10
S^{3}D-context	35.34	**22.09**	**4.48**	**22.11**

Table 4 Effects of using multiple temporal feature layers and span regression.

Span regression	conv5	conv6	conv7	conv8	conv9	conv10	mAP@0.5	# Spans
✓	✓	✓	✓	✓	✓	✓	**32.6**	252
	✓	✓	✓	✓	✓	✓	28.6	252
✓	✓	✓	✓	✓	✓		31.8	248
✓	✓	✓	✓	✓			30.7	240
✓	✓	✓	✓				27.6	224

Table 5 Effects of various design choices on S^{3}D performance, the span with ratio 1 is included by default. Each model is evaluated on THUMOS'14 when IoU threshold set to 0.5.

include 1.0 span	✓	✓	✓	✓
include 0.25 span		✓	✓	✓
include 0.5 span			✓	✓
include 0.75 span				✓
# Spans	63	126	189	252
mAP@0.5	27.5	29.5	31.1	**32.6**

Table 6 Activity detection speed during inference.

Method	FPS
S-CNN [29]	60
R-C3D [42] (on Titan X Maxwell)	569
R-C3D [42] (on Titan X Pascal)	1030
S^{3}D (ours on Titan X Pascal)	**1121**
S^{3}D (ours on GTX 1080 Ti)	**1271**

detection layers at different temporal levels by processing an untrimmed input video stream with enough temporal context.

ActivityNet. Table 3 shows our activity detection results on ActivityNet v1.3 validation subset along with state-of-the-art methods published recently. The proposed framework, using a single model instead of an ensemble, is able to achieve an average mAP of 22.11 that tops all other methods and perform well at high IoU thresholds, *i.e.*, 0.75 and 0.95, demonstrating that our model is able to predict precise temporal boundaries. Note that both TCN [7] and Chao *et al.* [6] are able to process an untrimmed video with deep networks and have considered temporal context features explicitly. However, their implementation is based on a two-stage approach where temporal proposals and proposal classification are optimized separately. The superior performance of our S^{3}D demonstrates the effectiveness of single-shot, end-to-end temporal activity detection, and our model also benefits from temporal contextual modeling.

6.4 Ablation Study

To understand S^{3}D better, we evaluate our network with different variants on THUMOS'14 to study their effects. For all experiments, we only change the certain part of the network and use the same evaluation settings. We compare the result of different variants using the mAP at IoU threshold 0.5. For a well-controlled experiments, we don't add temporal context features in all ablation experiments.

Default Span Ratio. As described in Section 3.3, by default we use 4 default spans per each temporal location. If we remove the spans with ratio 0.75, the mAP drops by 1.5%. By further removing the spans with ratio 0.25 and 0.5, the mAP drops another 3.6%. By only keeping the span with ratio 1.0, our model already has a strong performance (mAP 27.5%) since it already covers most ground truth instances in the dataset. Using a variety of default ratios make the task of predicting spans easier for the network and result in better performance.

Span Regression. The default spans are defined at fixed temporal locations. In order to generate precise predictions for starting and ending time of each activity instance, we adjust each default span by applying a temporal offset described in Equation 3. This technique, which we call span regression, allows our model to predict temporal spans at much smaller granularities. As shown in Table 4, span regression improves the mAP from 28.6% to 32.6%.

Multi-scale Default Spans. A major advantage of S^{3}D is using default spans of different scales on different temporal feature layers. To measure the advantage gained, we progressively remove layers and compare results. Table 4 shows a decrease in accuracy with fewer layers, dropping monotonically from 32.6% to 27.6%. This is because that different layers are responsible for predicting temporal activities at different lengths, which reinforces the message that it is critical to spread spans of different scales over different layers.

Activity Detection Speed. Since our model has a single-shot, end-to-end design with simple Conv3D building blocks, it is also very efficient. We compare the detection speed of our model with other state-of-the-art methods. The results are shown in Table 6. S-CNN [29] uses a sliding window proposal strategy and predicts

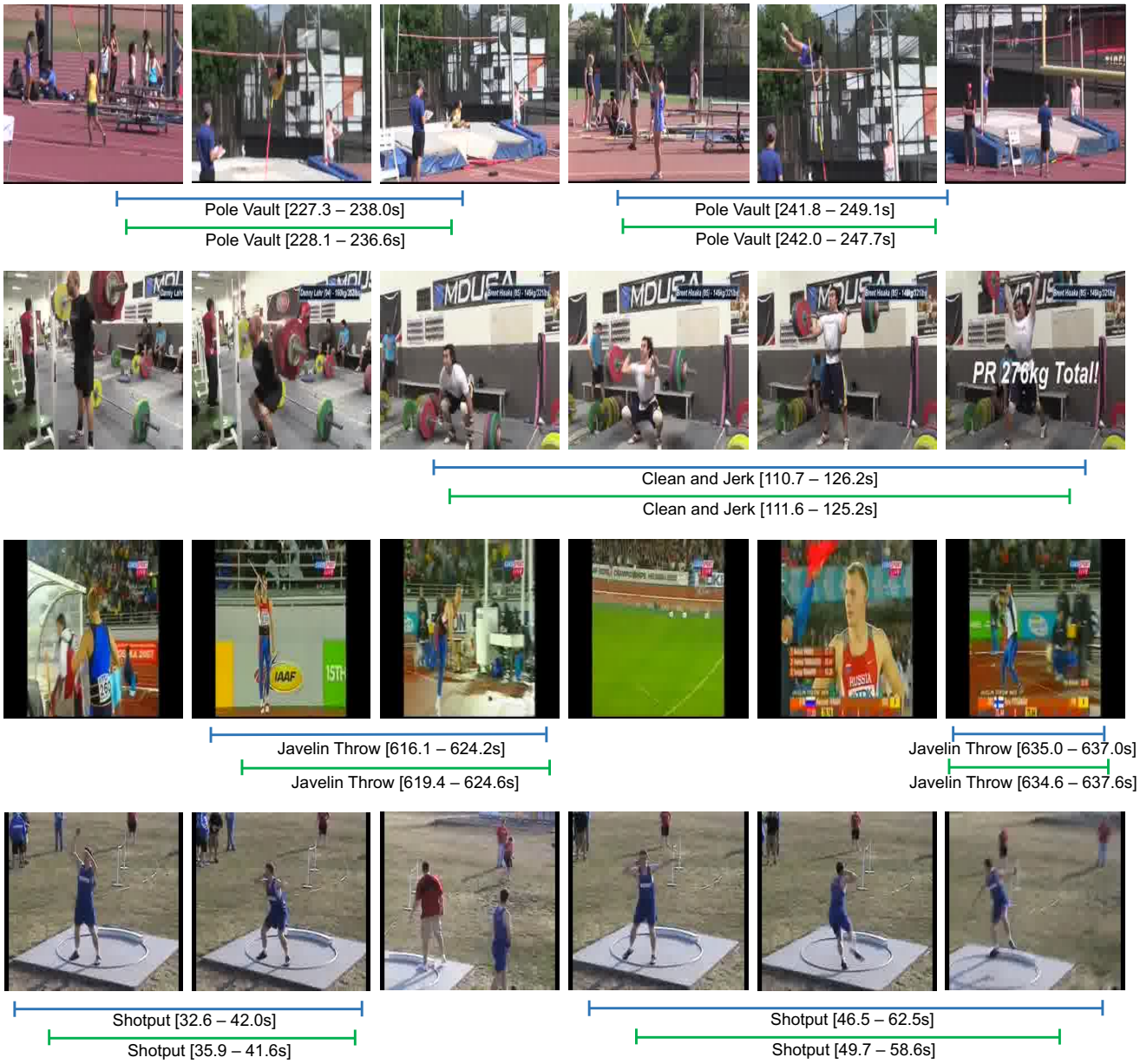

Fig. 5 Qualitative visualization of the top detected activities by S^{3}D (best viewed in color) on four different activity categories in THUMOS'14 dataset: *Pole Vault*, *Clean and Jerk*, *Javelin Throw* and *Shotput*. Each sequence consists of the ground-truth (blue) and predicted (green) activity segments and class labels.

at 60 FPS. R-C3D [42] constructs the proposal and classification pipeline in an end-to-end fashion using a Conv3D network and runs at 1030 FPS. Our framework applies the same idea to accept a wide range of frames as input and apply efficient Conv3D filters. However, we further boost the efficiency by eliminating explicit proposal generation and resampling step. For S^{3}D, the speed of execution is 1271 FPS on a single GTX 1080 Ti GPU, making it the fastest activity detector to date.

Qualitative Results. We provide qualitative results on THUMOS'14 and ActivityNet to demonstrate the effectiveness and robustness of our proposed S^{3}D network. As shown in Fig. 5 and Fig. 6, different video streams contain very diversified background context and different activity instances vary a lot in temporal location and scale. S^{3}D is able to predict the accurate temporal span as well as the correct activity category, and it is also robust to detect multiple instances with varying lengths in a single video.

In Fig. 5, S^{3}D can distinguish activity with minor differences such as the normal weightlifting compared to *Clean and Jerk*. It is also capable of detecting the same

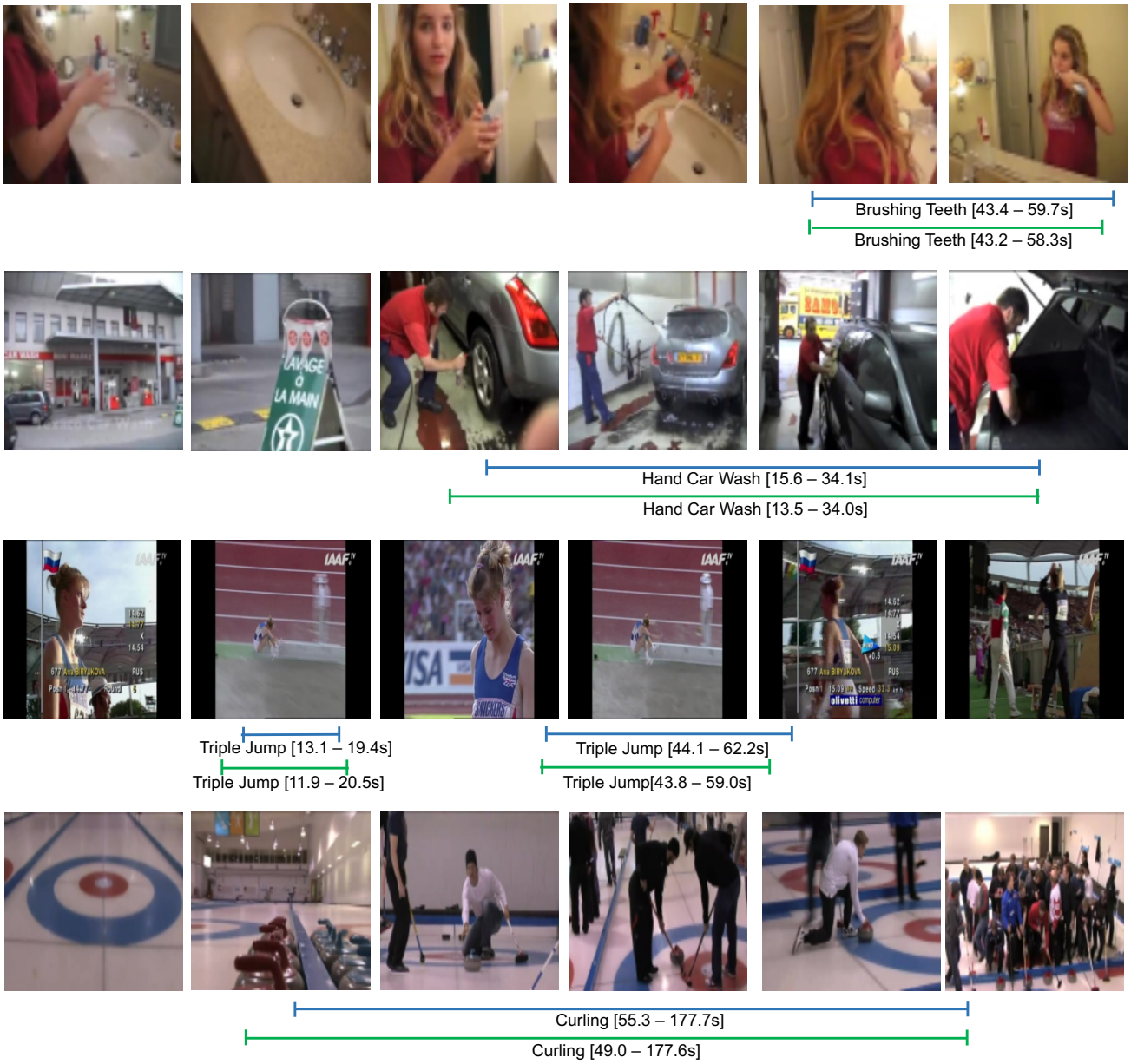

Fig. 6 Qualitative visualization of the top detected activities by S^{3}D (best viewed in color) on four different activity categories in ActivityNet dataset: *Brushing Teeth*, *Hand Car Wash*, *Triple Jump* and *Curling*. Each sequence consists of the ground-truth (blue) and predicted (green) activity segments and class labels.

activity sequence with different playing speed as shown in the *Shotput* example. In Fig. 6, S^{3}D is able to detect a variety of daily activities such as *Brushing Teeth* and *Hand Car Wash* whose temporal contexts differ a lot compared to sports activities. It is also robust to detect activities ranging from few seconds (*e.g. Triple Jump*) to few hundred seconds (*e.g. Curling*).

7 Conclusion

In this paper, we introduce S^{3}D, a Single-Shot multi-Span Detector for effective and efficient temporal activity detection. We design a simple network architecture by using only a fully Conv3D network on top of the raw video frames to jointly predict the temporal boundaries as well as activity categories. A key feature of S^{3}D is the use of multi-scale temporal span outputs attached to multiple temporal feature maps. Moreover, each feature map is augmented by temporal context fea-

tures by appropriately fusing features at multiple levels. With this framework, we achieved state-of-the-art performance on THUMOS'14 and ActivityNet benchmark datasets, while being efficient to run much faster than real time on a single GPU.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI, vol. 16, pp. 265–283 (2016)
2. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.: End-to-end, single-stream temporal action detection in untrimmed videos. In: Proceedings of the British Machine Vision Conference (BMVC) (2017)
3. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: Sst: Single-stream temporal action proposals. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pp. 6373–6382. IEEE (2017)
4. Caba Heilbron, F., Carlos Niebles, J., Ghanem, B.: 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, pp. 1914–1923 (2016)
5. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4733. IEEE (2017)
6. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1130–1139 (2018)
7. Dai, X., Singh, B., Zhang, G., Davis, L.S., Qiu Chen, Y.: Temporal context network for activity localization in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5793–5802 (2017)
8. Fabian Caba Heilbron Victor Escorcia, B.G., 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, pp. 961–970 (2015)
9. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
10. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization of actions with actoms. IEEE transactions on pattern analysis and machine intelligence **35**(11), 2782–2795 (2013)
11. Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. In: Proceedings of the British Machine Vision Conference (BMVC) (2017)
12. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448 (2015)
13. 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, pp. 580–587 (2014)
14. 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, pp. 770–778 (2016)
15. Jain, M., Van Gemert, J., Jégou, H., Bouthemy, P., Snoek, C.G.: Action localization with tubelets from motion. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 740–747 (2014)
16. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 3304–3311. IEEE (2010)
17. Jiang, Y., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes (2014)
18. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
20. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: Proceedings of the 2017 ACM on Multimedia Conference, pp. 988–996. ACM (2017)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision, pp. 21–37. Springer (2016)
22. Mettes, P., van Gemert, J.C., Cappallo, S., Mensink, T., Snoek, C.G.: Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 427–434. ACM (2015)
23. Oneata, D., Verbeek, J., Schmid, C.: Action and event recognition with fisher vectors on a compact feature set. In: Proceedings of the IEEE international conference on computer vision, pp. 1817–1824 (2013)
24. Oneata, D., Verbeek, J., Schmid, C.: The lear submission at thumos 2014 (2014)
25. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. Computer Vision–ECCV 2010 pp. 143–156 (2010)
26. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5534–5542. IEEE (2017)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91–99 (2015)
28. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: CVPR (2017)
29. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1049–1058 (2016)
30. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp. 568–576 (2014)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
32. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild (2012)

33. Tang, K., Yao, B., Fei-Fei, L., Koller, D.: Combining the right features for complex event recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2696–2703 (2013)
34. 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, pp. 4489–4497 (2015)
35. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6450–6459 (2018)
36. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 3169–3176. IEEE (2011)
37. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision, pp. 3551–3558 (2013)
38. Wang, L., Qiao, Y., Tang, X.: Action recognition and detection by combining motion and appearance features. THUMOS14 Action Recognition Challenge **1**(2), 2 (2014)
39. Wang, L., Qiao, Y., Tang, X., Van Gool, L.: Actionness estimation using hybrid fully convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2708–2717 (2016)
40. Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards good practices for very deep two-stream convnets. arXiv preprint arXiv:1507.02159 (2015)
41. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. CVPR (2018)
42. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: Proceedings of the International Conference on Computer Vision (ICCV) (2017)
43. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2678–2687 (2016)
44. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
45. Zhang, D., Dai, X., Wang, X., Wang, Y.F.: S3d: Single shot multi-span detector via fully 3d convolutional network. In: Proceedings of the British Machine Vision Conference (BMVC) (2018)
46. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: The IEEE International Conference on Computer Vision (ICCV), vol. 8 (2017)