Instance-aware Detailed Action Labeling in Videos

Hongtao Yang[1] Xuming He[2] Fatih Porikli[1]
[1]The Australian National University [2] ShanghaiTech University
{u5226028, xuming.he, fatih.porikli}@anu.edu.au

Abstract

We address the problem of detailed sequence labeling of complex activities in videos, which aims to assign an action label to every frame. Previous work typically focus on predicting action class labels for each frame in a sequence without reasoning action instances. However, such category-level labeling is inefficient in encoding the global constraints at the action instance level and tends to produce inconsistent results.

In this work we consider a fusion approach that exploits the synergy between action detection and sequence labeling for complex activities. To this end, we propose an instance-aware sequence labeling method that utilizes the cues from action instance detection. In particular, we design an LSTM-based fusion network that integrates frame-wise action labeling and action instance prediction to produce a final consistent labeling. To evaluate our method, we create a large-scale RGBD video dataset on gym activities for sequence labeling and action detection called GADD. The experimental results on GADD dataset show that our method outperforms all the state-of-the-art methods consistently in terms of labeling accuracy.

1. Introduction

Understanding complex activities from videos is a fundamental problem in computer vision and has a wide range of applications in surveillance, human-computer interaction and personal robotics [9]. Much progress has been made in the key tasks involved in activity analysis, including action classification [34, 12, 1, 29, 32, 4, 39], detection [22, 41, 28, 44], sequence labeling [40, 30, 18, 27, 16] and activity prediction [13, 14], etc. In addition, recent advances in depth sensors (e.g., Kinect) enable us to efficiently obtaining dense 3D measurements of dynamic environments, especially for the indoor setting. Exploiting such multi-modal videos for activity understanding has attracted much attention as they provide rich information about the object poses and shapes, and are less sensitive to object appearance and lighting condition [36, 23, 20, 25, 17].

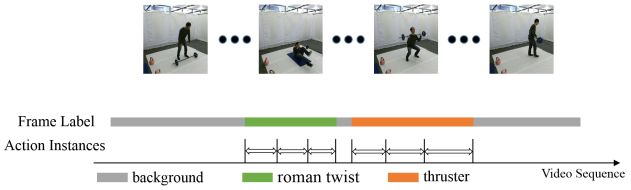

Figure 1. Dense sequence labeling & Detection. Each frame has an action class label, a set of consecutive frames forms an action instance. Note that several action instances can repeat right after each other.

Despite the progress in video-based action recognition, it remains a challenging task to fully parse complex activities that consists of a sequence of different actions in long video sequences. A key step towards understanding such activities is to produce a detailed frame-wise action labeling, which receives much less attention than other tasks in activity understanding (e.g., action classification or detection). Previous work typically focus on predicting action class labels for each frame without explicitly reasoning action instances in the sequence [40, 17, 16, 44, 27]. Nevertheless, such category-level frame-wise labeling strategy is limited in encoding more global constraints at the action instance level, which tends to produce inconsistent frame-level labeling for the entire sequence.

In this work, we propose to incorporate action instance information into detailed action labeling of complex activities in long videos by fusing the frame-level and instance-level action predictions. Here sequence labeling is performed in an online fashion, with prediction being made based on local observations and sequence history, while detection seeks action instances based on the spatio-temporal patterns of the full actions. As they capture complementary information at both action and frame level, our approach exploits the synergy between these two subtasks, and can thus handle arbitrary long videos with consistent performance in parsing multiple action instances. We also note that action classification techniques are not suitable for labeling tasks because in labeling, the total video length and the action duration are unknown.

We design a deep neural network pipeline using both

RNN and CNN to perform frame-level action labeling for complex videos with multi-class action instances. Our network consists of two interdependent modules for detection and labeling respectively. The action labeling module is a recurrent network based on the long short-term memory (LSTMs) [4], while the detection is performed by a stack-frame CNN [12], which takes a set of sampled frames and predicts action scores for every generated action proposal. To achieve consistent labeling, we integrate the predictions from the action detection module with an LSTM network for a refined sequence labeling results. Both the networks are built on a short-term video representation using dynamic images [1, 5].

To evaluate our method, we build a new large scale RGBD video dataset of gym exercise activities. It is specifically designed for sequence labeling and action detection with dense frame level labels. Unlike existing RGBD action recognition or detection datasets, our videos include more complex activities, typically defined by a continuous sequence of actions. For each activity, we simultaneously collect depth videos from four different viewpoints so that our evaluation consists of video data from multiple views. We refer to this dataset as Gym Activity RGBD Dataset (GADD). We extensively test our approach on the GADD dataset and compare with several baseline methods. Our network outperforms all the baseline methods and improves the performances of both subtasks.

Our main contributions are two-fold: First, we present a new dataset for activity recognition, detection and dense frame labeling in complex videos; Second, we develop a novel fusion method that integrates the subtasks of action instance detection and frame-level action labeling, which results in better overall performance for both tasks.

2. Related Work

Action Detection and Labeling Most action detection approaches aim to localize instances of a specific action class based on classifying generated action proposals. For example, [19] combines dense trajectories and frame level CNN features to detect actions from sliding window proposals. Such strategies have been improved by effective methods of generating high quality action proposals as in [43]. To capture temporal dynamics, LSTM networks have been widely used in representing action instances. Singh and Shao [30] propose an LSTM on top of a multi-stream CNN to model the dynamics for each proposal. Yeung *et al*. [41] use LSTM and reinforcement learning technique to progressively observe the video and refine its prediction on where to look next and when to make a decision. Other methods use structured representations to model the details of action instances. Yuan *et al*. [44] improves detection accuracy by explicitly finding the start, middle and end key frames of an action. Actions are commonly accompanied by objects, [22] build on this idea and propose detecting interactional objects and body parts using an object parsing network. They then extract motion features like HOF and trajectories on the parsed segments. RGB-D information has also been used in action detection literature [20, 25]. [20] extracts 3D spatial-temporal context of the actions using both gray scale and depth images. Shahroudy *et al*. [25] extract human body part from depth data and propose a part-aware LSTM network for recognition and detection. However, it is challenging for those methods to produce a consistent parsing of long videos with consecutive actions from multiple classes.

A more detailed approach to understanding complex activities in videos is through sequence labeling. One key challenge here is to achieve temporal consistency in the labeling. To better model temporal dynamics, Yeung *et al*. [40] propose an attention based MultiLSTM model with multi-label loss that intelligently select input frames and produce multiple outputs for a range of frames. [27, 16] utilize temporal convolution features and deconvolution operations to extract high level temporal dynamics for end-to-end sequence labeling. Lillo *et al*. [17] decompose a complex action into atomic actions and propose a pose based method to detect atomic actions. Ma *et al*. [18] designed a novel ranking loss to enforce the monotonicity of prediction scores, which encodes the activity progression constraint. By contrast, we integrate the detection with sequence labeling to achieve more consistent results. In addition, our work focus on learning action labeling from densely labeled videos, while some recent work [10, 24] take unsupervised or weakly supervised approaches.

Action Labeling Datasets Most commonly used large-scale action datasets are designed for understanding only RGB videos [31, 15, 12, 2]. Although these datasets are large and contain lots of variations, typical videos in these datasets contain only one action. Such datasets are designed for action recognition tasks, rather than detection and labeling tasks. Our new GADD dataset, by contrast, consists of multi-view RGB-D videos, and each video contains at least 12 action instances, making it ideal for action detection and sequence labeling.

The early RGB-D datasets from Microsoft [42, 36, 37] have relatively low resolution and consist of simple actions. Recent RGB-D datasets include more complicated and challenging movements, such as CAD-120[13], Office Activity [38] and RGBD-HuDaAct [21]. Very recently, a new dataset aimed for data-driven algorithms called NTU RGB+D [25] is introduced. It is by far the largest dataset with 56880 RGBD sequences containing over 4 million frames. However, they are mostly for the recognition task and include only one or a few simple actions per video. KSCGR dataset [26] is a cooking RGBD dataset containing a sequence of actions. However, the dataset is relatively

small and focuses on specific actions involving only two arms. Most similar to our RGB-D dataset is PKU-MMD [3], which is a large scale multi-view dataset that contains about 20 action instances in each video. However the actions are relatively simple and posed, making the dataset less realistic and challenging. Therefore, we build our own RGB-D dataset for evaluating action detection and labeling in this work, which includes realistic and complex gym exercise and workout activities. A detailed comparison between different datasets are presented in Sec.4.

Action Recognition and Video Representation Recently, deep network based representations have been widely adopted in activity analysis of RGB videos[9]. Both [12] and [29] use stacked frames as input and relies on the CNN network itself to extract temporal information. Many others resorts to the LSTM network to learn an action representation [45, 4]. [33] modifies the LSTM internal gates and propose Derivative of States (DoS) to get a video representation that emphasizes the motion salience. In terms of representing temporal dynamics, it is natural to extend the powerful 2D CNN into 3D domain, by extending the input and every convolutional kernel to 3D. Following this idea, [11] and [32] propose 3D convolutional neural network. Another series of work [5, 6, 1] aims to summarize the complex temporal dynamics into one compact image, which can be used by 2D CNNs for recognition. [35] extends this idea to optical flow images, further improves recognition accuracy.

3. Instance Aware Action Labeling

We now address the problem of labeling in the complex videos that comprises a sequence of action instances from multiple action classes. We aim to integrate global information from action detection to achieve consistent labeling for parsing complex activities. To this end, we design an instance-aware action labeling system which consists of three components: 1) a frame-wise labeling LSTM, 2) an action detection system to acquire global information at instance level, and 3) a fusion network that integrate both local and global information for consistent sequence labeling. In this section, we first introduce the frame representation employed in our method, and then we describe each model component in detail.

3.1. Frame Representation

We adopt a local frame representation that encodes the short-term temporal dynamics for the frame-level labeling and detection tasks. Specifically, we first computes a dynamic image feature and a single-frame feature, and then fuse them by concatenating their CNN representations.

Dynamic Images We represent the short-term spatio-temporal patterns by exploring the concept of dynamic images [1] based on rank pooling [6]. The original dynamic

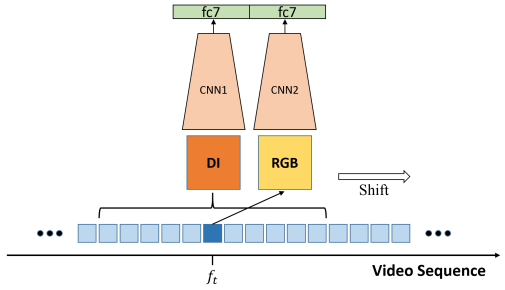

Figure 2. Representation for frame f_t. We use a window of n frames centered at the f_t to compute dynamic image. The dynamic image summarize the short term motion cue around f_t.

image (DI) aims to summarize the temporal evolution of a video with a single image representation.

In this work, we use an efficient approximation proposed by [1], which designs a fast rank pooling strategy. Concretely, the fast DI is computed as follows,

$$d = \sum_{t=1}^{T} [2(T-t+1) - (T+1)(H_T - H_{t-1})]\, x_t \quad (1)$$

where the coefficient $H_t = \sum_{\tau=1}^{t} \frac{1}{\tau}$. This fast approximation is essentially a weighted sum of the input frames. As in [1], we compute the dynamic images directly on the frame pixels, and the resulting d is a array of the same shape as the input frames features.

Figure 2 illustrates how we implement the dynamic images to get a frame representation. The DIs is applied to encode the local temporal windows centered at the video frames of interest. Our preliminary empirical study shows the DI is more effective to capture the short-term dynamics than being applied to long sequences. We therefore choose to use $n = 7$ frames centered at the frame f_t to construct the dynamic images at frame f_t. This way a dynamic image summarizes the local motions over a quarter of a second.

Single-frame Features We use slightly different single-frame features depending on the input is RGB or depth videos. For RGB videos, the single-frame feature is the raw RGB image. For depth videos, we also compute 3D surface normal map in addition to the raw depth frame. The normal vectors is able to better describe the surface shape [23]. Formally, the normal vector $\mathcal{N}$ at a point (x_0, y_0) on a surface $z = f(x, y)$ is given by

$$\mathcal{N} = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, -1]^T \quad (2)$$

We first apply median filter to the depth images, removing most of the noisy edges and filling in holes in the raw depth. Then we calculate numerical gradient by calculating finite difference on every point both horizontally and vertically,

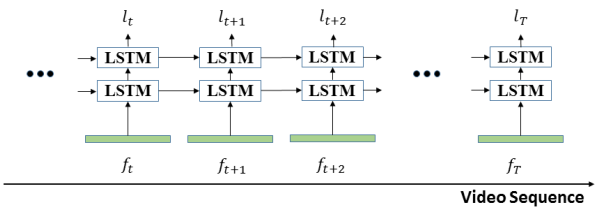

Figure 3. Overview of Sequence Labeling. We stack two layers of LSTM on top of the CNN feature to encode long-term temporal information.

obtaining a two-channel matrix corresponding to $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ respectively. We use the two channel matrix as the gradient normal map.

CNN-based Feature Fusion Given the dynamic image and single-frame features, we now develop a feature representation adopting a late fusion strategy that uses a two-stream CNN [29]. Instead of relying on optical flow which is noisy and expensive to compute, we use a stream of CNN to extract $fc7$ feature from the DI, which is more effective for capturing the short-term dynamics. A second stream of CNN computes the $fc7$ features from the single-frame feature, and we concatenate these two features together to represent the frame. Figure 2 gives a overview of the frame representation. It should be noted that to represent a video segment, we simply stack such frame representations as the input to CNN, as depicted in Figure 4

3.2. Frame-wise Action Labeling

To predict frame-wise action labels, we build an LSTM-based recurrent neural network on top of our frame representation. The RNN allows us to encode the long-term dynamics of the activities so that our prediction exploits both current and history observations.

Specifically, at each given time step, we first compute the local frame representation described in Section 3.1 as the input to the LSTM units. Our RNN consists of two layers of stacked LSTM with 512 hidden states, and the hidden representation of the top layer units are used to generate the probability of action classes through a softmax layer. We refer the reader to [8, 4] for the detailed equations of the LSTM units. Figure 3 shows an overview of the sequence labeling network.

To handle action labeling for general video sequences, we use the stateful LSTM to accommodate inputs that contain multiple instances and of variable lengths. The stateful LSTM accepts input of the current frames, but inherits the hidden state values from its previous iteration. This way, there is no need to specify the sequence length when initializing LSTM units. More importantly, in complex and long videos, a RNN with large temporal unroll step is not optimal because multiple actions with different patterns may be present within one temporal unroll and weights update. With stateful LSTM, the time unroll step can be set to be much smaller than the video length yet still capture long-term dynamics because of its state-inheritance nature. After all frames of a video are processed, the LSTM layer will reset its hidden state to make ready for the next video.

3.3. Action Detection with CNNs

We consider the task of action detection in the context of understanding complex activities, which typically involves a set of consecutive action instances from multiple classes. To efficiently localize these actions, we develop a two-stage detection method based on the Stack-frame CNN network [28]. Our detection pipeline takes a video as input, and first generates a pool of generic action proposals using an efficient CNN with binary output. We then learn a second CNN to classify the proposals into multiple classes. Both networks have the structure of stack-frame two-stream CNN that takes the stacked frame representation as input. The $fc7$ features of the two streams are concatenated and fed into a linear SVM for the final output. Figure 4 shows the overview of the detection module.

Action proposal generation. We reduce the search space for the action detection by first filtering out background and misaligned candidate windows. To achieve this, we build a CNN-based binary classifier, which is then applied to the input video sequence in a sliding window manner. The filter CNN predicts an actionness score for each window in its exhaustive search. The actionness score is the probability of the window being an action instance. Specifically, we evenly sample $k = 10$ frames and stack them together as the input to the CNN for each window.

We generate a pool of action proposals by removing the windows with probabilities lower than a threshold. We validate the threshold to strike a balance between speed and accuracy. In this work, it is empirically set to 0.8, which results in a 95% reduction in the number of windows and still maintains a recall rate of 1 with 0.5 IoU threshold.

Action instance classification. For each candidate window that passes the filter CNN, a multi-class CNN is used to predict the action label and its confidence scores. We further apply the non-maximum suppression (NMS) with a threshold of 0.3 to generate the final detection outcomes.

Concretely, we evenly select $k = 10$ frames from each sample window and stack the frame representations. Note that each dynamic image is computed using 5 consecutive frames centered at the frame of interest. By doing so, the actual frames used to obtain the video representation extends slightly beyond the windows itself, giving it extra context information [7].

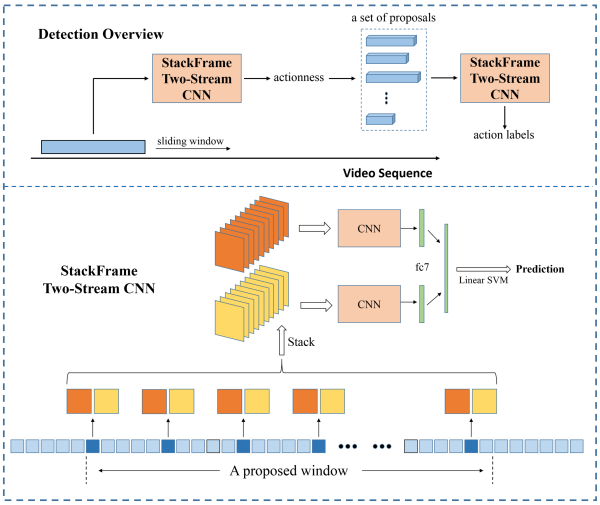

Figure 4. Overview of action detection module. We first compute the frame representation on 10 evenly sampled frames in the window. We then apply the two-stream stack-frame CNN. Note that in the process of obtaining dynamic images, frames outside the proposed window is used, which provides extra context information.

3.4. Instance-Aware Action Labeling Network

We now integrate the frame-wise action labeling module with the action detection module to achieve more consistent parsing of the input videos. The action labeling network captures rich context at the frame and category level while the detection module extracts instance-level information for the action units. A two-stage labeling process is used to exploit the synergy between those two modules.

Our first stage trains the detection module to generate a set of temporal bounding boxes, each box is associated with an action class label and a confidence score. We then use the detection results to compute a Frame Label Prior matrix (FLP). Specifically, let the detected actions of class l be a set of N_l temporal windows, denoted by $\{(s_i^l, e_i^l, c_i^l)\}_{i=1}^{N_l}$, where s_i^l, e_i^l are the start and end frame index, and c_i^l is the confidence score. For each action class, we select top $M = 10$ windows with highest confidence scores and sort them according to the scores. Denote the total number of action classes as L. At each time step t, we define a label prior matrix D_t of size M by L as follows,

$$D_t(m, l) = c_m^l \mathbb{1}(t \in [s_m^l, e_m^l]) \tag{3}$$

In the second stage, we design a two-branch fusion network to integrate the detection outcomes with the frame-wise action labeling task. Using the FLP matrix as input, we construct a convolution branch to encode the matrix into a feature vector containing the action instance information. Our convolution branch has two convolution layers followed by a fully connected layer with 64 output neurons. The output is concatenated with the hidden vector of the top-layer LSTM for the frame label prediction.

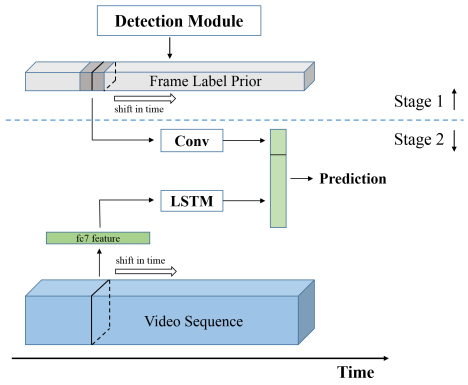

Figure 5. Two branch labeling network. The detection output around frame at time t (dark gray) is encoded using a small convolution network. The encoded instance cue is concatenated with the frame-level feature extracted by LSTM for prediction. Frame level features is computed according to Sec 3.1.

Figure 5 illustrates the structure of fusion network. During the prediction of frame label at time t, a slice of the array FLP around time t serves as input to a convolution branch. We jointly train the instance-aware labeling network end-to-end to obtain the final labeling prediction, and to ensure the CNN branch learns how to correctly extract action instance knowledges from FLP.

4. The GADD Dataset

We build a new RGBD video dataset for complex activity analysis, which is specifically designed to study the problem of sequence labeling and action detection. Unlike existing datasets that only contain one action per video, the RGBD videos in our dataset consist of a sequence of consecutive actions. Such scenarios are commonly seen in the real world problems and localizing actions becomes more challenging in this dataset. Figure 6 shows some sample frames from the GADD dataset. The dataset contain over 500 videos, each 1-2 minutes long. There are at least 12 action instances within each video. We now describe the details of the dataset.

Actions Our dataset contains 22 different action classes (23 with background class) of gym workout exercises (push up, KB swing, lunge twist, etc.). Four of these classes require the subject to use a tool, therefore human object interaction is introduced. The reason for choosing gym workout exercises is that all of them are well-defined actions and their starting and ending point can be easily determined, which is crucial for detection tasks where temporal localization is the main challenge.

Subjects There are in total 17 subjects participated in the video recording process. Different subjects chose different

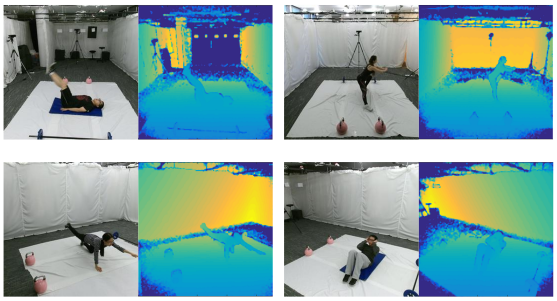

Figure 6. Sample frames from the GADD Dataset. These examples show the RGB and corresponding depth frames of 4 subjects performing 4 different exercises taken from 4 separate view points.

subset of the actions to perform. Once they have chosen, they perform all of them in a row, thus creating an action sequence.

View Points We set up cameras from 4 different angles on recording every videos. They are separated in a 180°space, at roughly about 0°, 60°, 120°and 180°respectively. Because the subject will be constantly moving, the angles relative to the subject will change during filming. The four different views are separated far enough, which creates large pose variation in the dataset. We assume the location of the cameras are unknown for our task.

Data Formats We collected the data using Microsoft Kinect v2, and recorded the RGB and depth maps for each frame. After filming all the actions, we manually label each actions by its start and end frame numbers. Each video sequence has three components: Depth frames, RGB frames and its action annotations.

For the depth channel, each frame is of resolution 512x424, with each pixel value representing the distance between a real world point and the kinect camera plane. The depth value is the unit of mm, and is quantized using 16 bit unsigned integer. For the RGB channel, we center crop each frame to 1080x1080 to retain its aspect ratio, and then resized to 270x270. Note that we do not use the RGB cues in this work. For the annotation, we record the annotated class and frame indices of every action instances.

Datasets Comparison We present a summary of existing public RGB(-D) datasets for action understanding and compare them with our GADD benchmark in Table 1. We can see that our video sequences have much more action instances per video than other datasets.

5. Experiments

We evaluate our instance-aware sequence labeling method on the RGB and depth videos in the GADD dataset. We first describe experimental setup and evaluation metrics

Datasets	Videos	Class	Subjects	view points	# actions per video	Modalities
Activity Net	20000	200	-	-	1.54	RGB
UCF-101	13320	101	-	-	1	RGB
HMDB51	7000	51	-	-	1	RGB
MSR-Action3D	567	20	10	1	1	D+3DJoints
CAD-60	60	12	4	-	1	RGB+D+3DJoints
RGBD-HuDaAct	1189	13	30	1	1	RGB+D
MSRDaily Activity3D	320	16	10	1	1	RGB+D+3DJoints
CAD-120	120	10+10	4	-	1	RGB+D+3DJoints
Office Activity	1180	20	10	3	1	RGB+D
NTU RGB+D	56880	60	40	80	1	RGB+D+IR+3DJoints
GADD	900	22	18	4	12 - 16	RGB+D

Table 1. Comparison between GADD dataset and some of the other popular available datasets for 3D video understanding.

in Sec. 5.1. Then, we report the results on sequence labeling with comparisons to several strong baseline methods, which is the focus of this paper. Finally, we show that not only detection can help improve labeling, but labeling can boost detection performance as well, indicating that action detection and sequence labeling are mutually beneficial.

5.1. Data Preparation and Evaluation

Dataset Split & Training details We split the dataset into two training sets of 300 (TR1) and 100 (TR2) videos respectively and a test set of 100 videos. For the training of the fusion network, we use TR1 training set to learn the detection system first and TR2 training set to train the instance-aware labeling model.

For the stack CNN in action detection system, we train the network on TR1 and adopt the Adam optimizer with base learning rate is 0.0001 and batch size 256. For the frame-wise LSTM labeling network, we use the SGD optimizer with a base learning rate 0.01 and batch size 1. This is also trained on TR1. Then we apply the trained detection CNN and frame-wise LSTM on TR2 to obtain the detection and labeling results, which are used to train the instance-aware labeling net on TR2. For this network, we use the Adam optimizer with base learning rate 0.00001 and batch size 1.

Evaluation metrics We employ two different evaluation metrics for the task of sequence labeling. One is *accuracy* which reflects the frame level performance (how many frames are correctly labeled). The other is *f-score* which is calculated with *precision* and *recall*. The *f-score* reflects the class specific instance level performance.

For the task of action detection, we compute the intersection-over-union (IoU) between the predicted temporal windows and the ground truth and consider the detection is correct if IoU>0.5. We then use the average precision to report the results on each class and the mean average precision (mAP) for overall performance.

	Accuracy(%)	f score
CNN	76.32	0.72
LSTM	78.45	0.74
Attention LSTM[40]	80.27	0.74
Bi-LSTM[30]	79.88	0.75
RankingLoss LSTM[18]	80.75	0.76
LSTM (with smoothing)	80.33	0.73
LSTM (instance-aware)	**83.78**	**0.81**

Table 2. Sequence labeling accuracy and F-score fused with detection results for RGB videos. Noticeable improvement can be seen in both evaluation metrics.

	Accuracy(%)	f score
CNN	78.86	0.69
LSTM	81.25	0.78
Attention LSTM[40]	81.36	0.77
Bi-LSTM[30]	81.87	0.79
RankingLoss LSTM[18]	82.03	0.79
LSTM (with smoothing)	82.13	0.78
LSTM (instance-aware)	**84.56**	**0.86**

Table 3. Sequence labeling accuracy and F-score fused with detection results for Depth videos. Noticeable improvement can be seen in both evaluation metrics.

5.2. Results on Sequence Labeling

Action Labeling We show the results of sequence labeling in Table 2 and Table 3 for RGB and depth video respectively. We compare the performance of the basic LSTM model, three state-of-the-art methods [40, 30, 18] and our fusion LSTM model based on the same video representation. To put our instance-aware fusion method in context, we also implemented naive smoothing method (majority voting) for comparison.

First, we note that by modeling the long-term dynamics with the basic LSTM, we witness significant improvements compared to the traditional CNN. Attention LSTM performs slightly better than basic LSTM, thanks to its attention mechanism that is capable of focusing on relevant frames and capturing longer-range dynamics. By introducing temporal consistency constraints, [30, 18] also achieve slightly better performances. However, our instance-aware LSTM network fused with detection can effectively model global dynamics, outperforming the other state-of-art methods consistently. The fusion of detection and sequence labeling generates a noticeable improvement in both accuracy and F-score, indicating the synergy between action detection and sequence labeling. Moreover, comparing the last two rows in the table, we can see moderate improvements on accuracy but large jumps in f-score. This clearly indicates that our instance-aware labeling method works more effectively for action instances, while naive smoothing method treats every frames equally. Although smoothing operation can improve accuracy, most improvements is happening within background frames. When applying such operation on complex sequences, it will blur the instance boundaries, leading to no improvements or even worse results in f-score.

Figure 7 visually shows the labeling results with and without the use of LSTM. Figure 8 gives a detailed class-by-class comparison on the *f score* induced by the fusion of action detection. We can see that our joint method outperforms the basic LSTM across all the action classes.

Concretely, we can make several observations from Figure 7. First, by comparing the third row to the second, we can see that the basic LSTM mainly corrects the short chunks of false prediction inside an action, which produces a smooth labeling outcomes. Second, from the forth row, we observe that the behavior of attention LSTM has large variations: while it does capture longer-term dynamics, it also seems to attend to irrelevant frames. In contrast, as shown by the difference between the third and fifth row, by fusing the detection results, our method effectively corrects false predictions within the interior of action instances, as well as during the transition between non-action (background) and action, where the frame features can be ambiguous. Furthermore, in comparison with the sixth row, it is evident that the proposed instance-aware method can handle false predictions more adaptively. Naive smoothing methods are ineffective when a set of consecutive frames are predicted falsely.

The performance improvement of our method is likely due to the following reason. Within a complexity action sequence, the frame level prediction inferred by short term dynamics can easily make mistakes as the basic movements of human limbs that form a complex action are likely to appear in some other actions, which causes confusion with local information only. The problem can be alleviated by joint prediction with more global contextual information from sequence history and action instances. What's more, the frame labels during the transition between non-action and action are ambiguous in nature. If a correctly detected window has high IoU for an action instance, it can facilitate labeling of the frames during transition. However, it is noteworthy that combining detection results of poor quality can sometimes lead to deteriorated results, as shown in *Video 3* in Figure 7.

5.3. Impact on Action Detection

While our fusion method significantly improves labeling performance, we also found the labeling outcomes can help boost detection performance. Table 4 shows the mAP scores and after re-weighting the detection score with labeling results, we can see that there is a significant improvement in the overall mAP. When the detection system localizes the

Figure 7. Comparing different sequence labeling results. *Black* is background, *Green* is an action, and *Red* is false prediction. **First row**: ground truth. **Second row**: labeling results with CNN only. **Third row**: labeling results with added LSTM. **Forth row**: attention LSTM labeling results. **Fifth row**: instance-aware labeling results. **Sixth row**: Smoothing applied on basic LSTM results. *Video 1* and *Video 2* illustrate how joint prediction can help improve accuracy by considering global information at instance level. *Video 3* shows a failure case due to poor-quality detections.

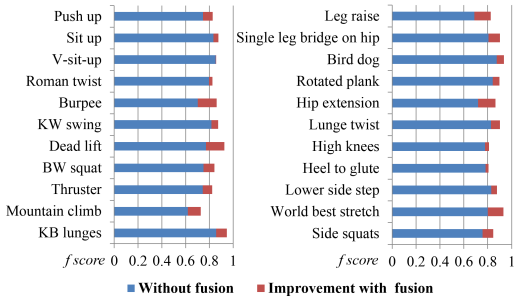

Figure 8. Per-class F-score of sequence labeling. With fusion with sequence labeling, we witness performance gain on all the action classes.

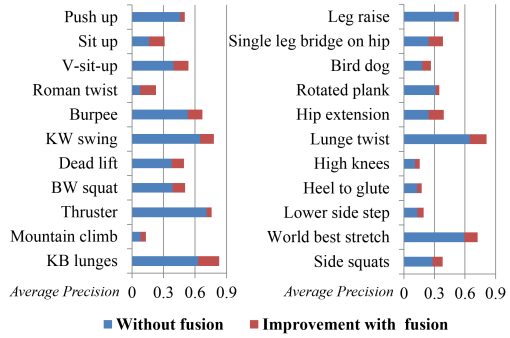

Figure 9. Per-class average precision of action detection. By the fusion with sequence labeling, we witness performance gain on all the action classes.

action instances, their class labels can be noisy due to lack of long-term temporal context. On the other hand, the action labeling can provide more stable results regarding the action class for each frame. By fusing them together, we are able to reduce the false positives from incorrect action classes and achieve better performance.

Figure 9 shows detailed class-by-class improvement on the *average precision* induced by the fusion of sequence labeling. Figure 10 shows the typical precision recall curves

Video modality	mAP(%)	mAP with SL
RGB	39.98	44.38
Depth	35.59	45.81

Table 4. Mean Average Precision for action detection for RGB and depth videos. The IoU threshold is set to 0.5 during evaluation.

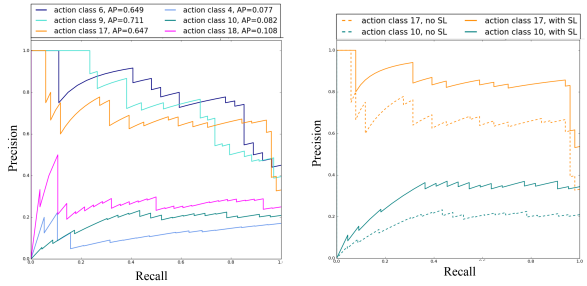

Figure 10. Precision Recall Curve. **Left**: The graph shows three action classes with high-quality detection and three classes with low-quality detection. **Right**: The graph shows the effect of sequence labeling fusion on detection for two action classes.

of detection from several action classes. The left panel shows three action classes with highest PR curves and the three with lowest ones. The right panel demonstrates the improvement from fusion for two typical action classes, which is evident.

6. Conclusion

In this work we have presented a instance-aware labeling approach for dense activities labeling in complex videos, which combines the frame level information from LSTM with instance level information from action detection. We explore the synergy between action detection and sequence labeling. We design an LSTM + CNN fusion network to jointly solve the problem, leading to consistent improvements over strong baselines methods. We also build a new large-scale RGBD dataset for complex activity understanding, called GADD.

References

[1] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *CVPR*, 2016.

[2] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[3] L. Chunhui, H. Yueyu, L. Yanghao, S. Sijie, and L. Jiaying. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017.

[4] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, June 2015.

[5] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, pages 5378–5387, 2015.

[6] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, pages 5378–5387, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[8] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[9] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *arXiv preprint arXiv:1605.04988*, 2016.

[10] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, pages 137–153. Springer, 2016.

[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(1):221–231, 2013.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[13] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[14] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2016.

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[16] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. 2017.

[17] I. Lillo, J. Carlos Niebles, and A. Soto. A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets. In *CVPR*, pages 1981–1990, 2016.

[18] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *CVPR*, pages 1942–1950, 2016.

[19] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *CVPR*, pages 756–763, 2014.

[20] B. Ni, Y. Pei, P. Moulin, and S. Yan. Multilevel depth and image fusion for human activity detection. *IEEE transactions on cybernetics*, 43(5):1383–1394, 2013.

[21] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *Consumer Depth Cameras for Computer Vision*, pages 193–208. Springer, 2013.

[22] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine grained action detection. In *CVPR*, pages 1020–1028, 2016.

[23] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *CVPR*, pages 716–723, 2013.

[24] K. Papoutsakis, C. Panagiotakis, and A. A. Argyros. Temporal action co-segmentation in 3d motion capture data and videos. 2017.

[25] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. *arXiv preprint arXiv:1604.02808*, 2016.

[26] A. Shimada, K. Kondo, D. Deguchi, G. Morin, and H. Stern. Kitchen scene context based gesture recognition: A contest in icpr2012. In *Advances in depth image analysis and applications*, pages 168–185. Springer, 2013.

[27] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. 2017.

[28] Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns.

[29] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[30] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1961–1970, 2016.

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

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[33] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *ICCV*, pages 4041–4049, 2015.

[34] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[35] J. Wang, A. Cherian, and F. Porikli. Ordered pooling of optical flow sequences for action recognition. *arXiv preprint arXiv:1701.03246*, 2017.

[36] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *ECCV*, pages 872–885. Springer, 2012.

[37] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297. IEEE, 2012.

[38] K. Wang, X. Wang, L. Lin, M. Wang, and W. Zuo. 3d human activity recognition with reconfigurable convolutional neural networks. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 97–106. ACM, 2014.

[39] Y. Wang, M. Long, J. Wang, and P. S. Yu. Spatiotemporal pyramid network for video action recognition. 2017.

[40] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738*, 2015.

[41] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *arXiv preprint arXiv:1511.06984*, 2015.

[42] G. Yu, Z. Liu, and J. Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *ACCV*, pages 50–65. Springer, 2014.

[43] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, pages 1302–1311, 2015.

[44] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng. Temporal action localization by structured maximal sums. 2017.

[45] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015.