Computer Vision and Image Understanding 175 (2018) 24–31

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Back to the beginning: Starting point detection for early recognition of ongoing human actions

Boyu Wang [*], Minh Hoai

Stony Brook University, Stony Brook, NY, 11794, USA

ARTICLE INFO

Communicated by Nikos Paragios

MSC:
41A05
41A10
65D05
65D17

Keywords:
Action early recognition
Online action detection
Event detection

ABSTRACT

We address the task of recognizing the category of an ongoing human action from a video stream. This task is challenging because of the need to output categorization decisions based on partial evidence—the action has not finished and not all information about the action has been observed. This task is further complicated because the ongoing action is submerged in the stream of data and the start of the action is not given. Existing methods for early recognition usually ignore this issue, making unrealistic assumption about the availability of the starting point of the ongoing action. In this paper, we prove the importance of starting point detection and subsequently propose a method to determine the start of an ongoing action. Our method is based on a bidirectional recurrent neural network that computes the probability of a frame to be the starting point by comparing the dynamics of the actions before and after the frame. Experiments on three datasets show that our method can reliably detect the starting point of an ongoing action, improving the early recognition accuracy.

1. Introduction

The task we study here is early recognition, which aims to detect and recognize ongoing human actions from a video stream as soon as possible, as illustrated in Fig. 1. This task arises in many situations, and the ability to make early and reliable decisions is the key to enable applications in a wide range of fields, from robotics and entertainment to surveillance and health care.

Many methods have been developed for human action recognition (Yacoob and Black, 1999; Oliver et al., 2004; Wang and Hoai, 2018b, 2016), but most of them focus on improving the accuracy of offline processing rather than the timeliness of the decision making. Existing action recognition algorithms have a limitation in processing sequential data as they are only trained to recognize complete actions, once the actions have finished and all information about them is obtained. But for early recognition, it is necessary to have the ability to recognize the categories of the partial actions. Partial actions, however, are ignored in the training process of most existing action recognition algorithms. Only in the last few years have there been methods (Ryoo, 2011; Ryoo et al., 2015; Hoai and De la Torre, 2014; Kong et al., 2014; Cao et al., 2013; Huang et al., 2014; Kong and Fu, 2015; Xu et al., 2015; Raptis and Sigal, 2013; Hu et al., 2016; Ellis et al., 2013; Zanfir et al., 2013; Lan et al., 2014; Kitani et al., 2012; Vondrick et al., 2016; Yu et al., 2015; Li et al., 2016; Ma et al., 2016; Singh et al., 2017; Soomro et al., 2016; Wang and Hoai, 2018a; Shou et al., 2018) that learn temporal models for partial actions. However, many existing methods for early recognition make unrealistic assumptions about the detection process—they assume that an ongoing action can be easily identified and separated from the video stream. Some methods assume that the start of a human action is known, so it is sufficient to focus on learning a good classifier for partial actions. Some methods even assume the observational ratio of an ongoing action is known. The observational ratio is the proportion of the action that has been observed at the time of making the decision, and it is only known if the start and the duration of the action are known.

Some methods do not require the observational ratio of an ongoing action to be known, e.g., (Hoai and De la Torre, 2014; Raptis and Sigal, 2013; Huang et al., 2014; Lan et al., 2014; Ryoo et al., 2015; Kong and Fu, 2015; Yu et al., 2015; Zanfir et al., 2013), but they assume the computational model of partial actions can be used to localize the start of the action. One simple approach is to use the sliding window technique: the action classifier is used to evaluate multiple video segments that correspond to different possible starting points of the action, and the segment with the highest classification confidence is considered as the location of the ongoing action. This approach, however, does not work well in practice, as will be shown in our experiments. This is because the computational models for partial actions are normally trained to optimize the classification accuracy, not the localization accuracy. As such, using the classification confidence to localize an ongoing action yields poor performance, especially when there are multiple action classes.

[*] Corresponding author.
E-mail address: boywang@cs.stonybrook.edu (B. Wang).

https://doi.org/10.1016/j.cviu.2018.10.001
Received 30 January 2018; Received in revised form 20 July 2018; Accepted 14 October 2018
Available online 25 October 2018

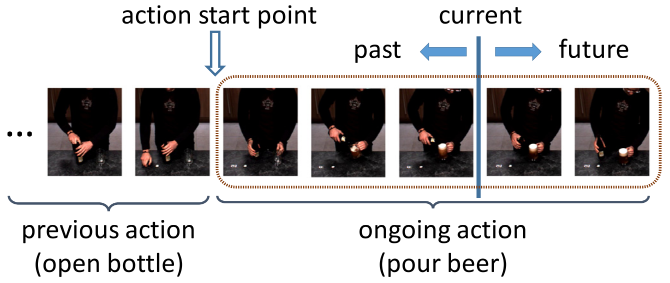


Fig. 1. How can we recognize the category of an ongoing action that is submerged in the stream of data? We propose a method to automatically determine the starting point of the ongoing action to improve the recognition performance.

There exist temporal models such as Hidden Markov Models (Rabiner, 1989; Kitani et al., 2012) and Recurrent Neural Networks (Rumelhart et al., 1986; Hochreiter and Schmidhuber, 1997; Vondrick et al., 2016) that can theoretically recognize an action without explicitly estimating the start of the action. These models use a state vector to store the integrated information about an ongoing action. At each time step, the state vector is updated given a new video frame, and the stored information in the state vector is be used to make the classification decision. However, the state vector is designed to incorporate all past sensor observations, which dilutes the subtle signal at the onset of a human action. As a result, a state-based model may be slow in recognizing an ongoing action.

Our first contribution in this paper is a set of experiments that prove the importance of estimating the start of the action for early recognition. This applies to various computational models, including segment-based and state-based models. Our experiments also reveal the poor performance of the sliding window approach for localizing the start of the action.

Our second contribution in this paper is the development of a novel method to estimate the start of the ongoing action, as shown in Fig. 1. Our method is based on Bidirectional Long-Short Term Memory (BLSTM) networks (Graves and Fernández, 2005). The network is trained to output a probability distribution for the location of the starting point. To train this network, we propose to use a novel loss function that is defined based on the difference between the cumulative distribution functions instead of the difference between the probability density functions.

The proposed method can estimate the starting point of an ongoing action with a small margin of errors. The median error is 18 frames, and the predicted starting point is within 5 frames of the actual starting point for 44.2% of the cases. Using the estimated starting point, we can improve the accuracy of all early recognition methods, even the one that is least sensitive to the location of the starting point.

2. Related works

To the best of our knowledge, no prior work has studied the benefits of estimating the starting point for early recognition. Most existing methods for early recognition either ignore the starting point, expect it to be given, or assume it can be reliably found using the sliding window approach.

What is being addressed here should not be confused with action detection (also known as action localization) or temporal segmentation (Ma et al., 2016; Yu and Yuan, 2015; Yuan et al., 2009; Tian et al., 2013; Lan et al., 2011; Ni et al., 2014; Dave et al., 2017; Hoai et al., 2011; Hoai and De la Torre, 2012; Hoai et al., 2014; Wei et al., 2018). Although action detection and temporal segmentation methods can determine the locations of the actions, they are designed for retrospective analyses in which the actions have finished and all information about the actions are observed. The problem being addressed here is more challenging; we need to determine the location of the starting point while the action is still going on. Furthermore, our ultimate goal is not precise localization; we only aim for an error margin that is tolerable by early recognition methods.

Our work is different from anomaly detection (Marchi et al., 2015; Malhotra et al., 2015; Kiran et al., 2018). Anomaly detection methods can be used for detecting abnormal events, but human actions are not abnormal. By definition, abnormal events are rare and they cannot be explained by the events that are normally observed; most existing abnormal event detection methods use some form of reconstruction/fitting error as the indicator for abnormality. But human actions are not abnormal, so anomaly detection methods are not applicable here.

Our work is related to but different from on-line change point detection in time series analysis (Basseville et al., 1993; Poor and Hadjiliadis, 2009; Picard, 1985). Change point detection methods work by detecting the locations where abrupt statistical changes occur. However, most existing methods for change point detection either scale poorly with the dimensionality of the time series data or assume the distribution of high dimensional data is known (Enikeeva and Harchaoui, 2013; Berkes et al., 2004; Xie et al., 2013). But human action data is high dimensional and the statistical distribution of the data is unknown and hard to estimate, so traditional change point detection algorithms are not suitable to detect the starting points of human actions. In this paper, instead of comparing the distributions of simple statistics, we propose to use Bidirectional LSTM to compare the non-linear dynamics of human actions before and after the change boundary. The Bidirectional LSTM can be trained with supervised learning, and it is an effective method for estimating the starting points of ongoing actions.

3. Benefits of knowing the start of an action

In this section, we consider several representative action recognition methods and evaluate their abilities to detect and recognize the category of an ongoing action. Our experiments reveal a large performance gap between knowing and not knowing the start of the action.

What we are about to present will seem to be obvious, but its important implication that has been overlooked by the research community. Many methods (Hu et al., 2016; Li et al., 2016; Ryoo and Aggarwal, 2009; Ryoo et al., 2015; Cao et al., 2013; Lan et al., 2014) for early recognition assume that the start of the ongoing action is known and it is sufficient to train a good computational model for *classifying* partial actions. Undoubtedly, the ability to correctly classify partial actions is an important subproblem of early recognition, but we prove here the importance of detecting the starting point. Not knowing the starting point and using a naive method to estimate it can severely degrade the performance of an action classifier; the extent of the severity is so big that it might negate the improvement obtained by having a better classifier. Thus the starting point of the ongoing action cannot be assumed to be known. This is the first paper that formally studies and raises this important issue, and this is one contribution of our paper.

3.1. Dataset

For the study of this section, we use the Montalbano Gesture dataset (Escalera et al., 2014). This dataset was captured with a Microsoft Kinect depth camera. In each sequence, each subject was recorded in front of the camera performing several natural communicative gestures. The gestures were performed by 27 different individuals under various conditions. The dataset contains 20 different actions. The dataset is divided into train, validation, and test subsets, and they contain 393, 287, and 276 full sequences respectively. The dataset comes with skeleton data, each frame containing (x, y, z) positions of 20 body joints, which are concatenated to create a 60-dimensional vector.

3.2. Action recognition methods

We consider several representative methods for ongoing action classification, developed based on state-of-the-art classifiers and temporal models: SVM (Vapnik, 1998), HMM (Rabiner, 1989), and LSTM (Hochreiter and Schmidhuber, 1997). These methods have different *recognition philosophies*. We implement and optimize each method based on its preferred representation of the input sequence, i.e., a feature representation that is commonly used and well suited for the method evaluated. Evaluation of all methods is of course carried out using identical data.

Segment-based SVM. We implement a method that is based on the state-of-the-art method for recognizing human actions using skeleton data (Luo et al., 2013). This method uses an SVM for classifying and computing the confidence of the classification decision. The input to the SVM is a temporal segment of a video. The method first computes a feature vector to represent the video segment, and subsequently feeds it to the SVM for classification. We refer to this method as Segmented-based SVM or SVM for short.

The input feature vectors are based on sparse coding (Yang et al., 2009; Lee et al., 2006) and temporal pyramid pooling (Luo et al., 2013). This type of feature has been shown to achieve state-of-the-art recognition performance with max-margin classifiers (Luo et al., 2013). First we learn a visual dictionary for skeleton data, and then use the dictionary to encode skeleton data at every time step. Given a training set $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N]$, where each $\mathbf{s}_i$ represents one skeleton pose, the visual dictionary can be learned by optimizing:

$$\min_{\mathbf{D}, \{\boldsymbol{\alpha}_i\}} \sum_{i=1}^{N} \left(||\mathbf{s}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda ||\boldsymbol{\alpha}_i||_1 \right), \tag{1}$$

where the matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_M]$ is the dictionary with M atoms and $\boldsymbol{\alpha}_i$ is a sparse vector of coefficients for encoding the training pose $\mathbf{s}_i$. Once the dictionary has been learned, it can be used to encode any pose vector $\mathbf{s}$ (not necessary part of the training set) by finding $\boldsymbol{\alpha}$ that minimizes: $\|\mathbf{s} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$. Once the sparse encoding vectors for every frame in a video segment have been obtained, the feature vector to represent the video segment is obtained using max pooling with a 3-layer temporal pyramid. This yields a feature vector with $7M$ dimensions, where M is the size of the visual dictionary $\mathbf{D}$.

To train an SVM that can recognize the categories of partial actions, we augment the training set to include partial actions at different observation ratios. To mediate the fact that the partial actions correspond to small observation ratios might be ambiguous to recognize, we use smaller weights for partial actions while training the SVM. Note that the need for modeling partial events has been proposed before (e.g., Hoai and De la Torre, 2014; Ryoo, 2011), and it is only necessary for early recognition of ongoing actions.

LSTM recurrent neural network. We implement a method for human action recognition based on the LSTM Recurrent Neural Network (Hochreiter and Schmidhuber, 1997). Interests in LSTM networks have grown with the success of deep learning (Weston et al., 2016), and they have successfully been used for human action recognition (Veeriah et al., 2015; Donahue et al., 2015; Yue-He. Ng et al., 2015).

We train a 3-layer LSTM network. At each time step, the input to the network is a 60-dimensional vector for the human pose (3D positions of 20 body joints) and the output is a probablity vector of length $C + 1$, where C is the number of action classes. There is a special class for non-action. The training data is a collection of multiple sequences of human actions. Each training sequence is not a short clip of a segmented human action. Each training sequence contains multiple human actions, where two actions might occur one after another or be sandwiched by a non-action sequence. Here, we train the LSTM network on long sequences to prepare for the testing scenarios where the network needs to recognizes an ongoing sequence in an unsegmented data stream. To train the LSTM network, we optimize the parameters of the network to minimize the sum of the cross-entropy losses at all time steps. Optimization is done using backpropagation through time. To avoid the prohibitive cost of backpropagation on long sequences, we only unroll the network with a fixed number of time steps. To preserve long-term context, we retain the hidden state of the last element in the previous sequence when transitioning to the next sequence.

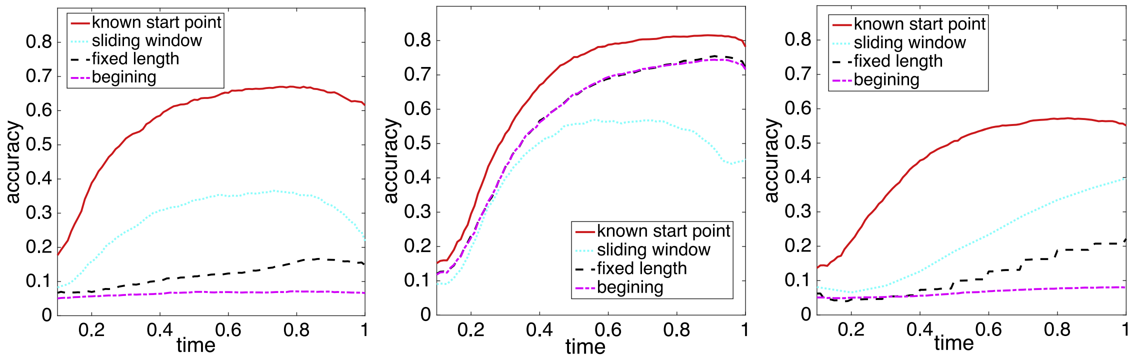
Hidden Markov model. We implement a method for human action recognition based on HMMs. For each action class (including the non-action class), we train an HMM with 6 hidden states, where each hidden state is parameterized by a mixture of 80 Gaussians with diagonal covariances. The number of Gaussians in a mixture model seems to be too large, but this provides better recognition performance than using a smaller number of Gaussians. Similar parameter settings (6 hidden states, 125 Gaussians) were also used for human action recognition (Xia et al., 2012). Once the set of HMMs have been trained, they can be used to predict the action class of a test sequence. The predicted class is the one with the largest posterior probability.

3.3. Experimental results

We evaluate the performance of SVM-based, LSTM-based, and HMM-based methods on the Montalbano dataset. We consider a realistic scenario when the start of the ongoing action is not given. In this case, there are several approaches that can be used with a given classification model. One approach is to consider multiple starting points and output the decision with the highest level of confidence. We refer to this approach as *sliding window*. Another approach is to ignore the starting point and evaluate the classifier using all the observed sensors values from the past. We refer to this approach as *beginning*. The third approach is to go back a fixed number of time steps and to use the observations since that time only—this method is referred to as *fixed length*. In our experiments, the number of time steps to look back is 80, which is the 95 percentile of the action length.

Fig. 2 shows the performance of the three methods using different approaches to deal with the unknown starting points of ongoing actions. In all figures, the horizontal axis shows the observational ratio, which is the proportion of the action that has already occurred at the time of making the classification decision. The vertical axis shows the accuracy of the classifier, averaged over all action and non-action subsequences in the long testing sequences of multiple actions. We also evaluate the methods for the ideal case when the starts of ongoing actions are given (referred as *known start point*). Not surprisingly, all methods achieve their best performance when the starts of the ongoing actions are known. The performance gap between knowing and not knowing the starting points is huge. Even for the LSTM method where several performance curves appear to be close, the performance gap is actually big. For example, at 70% detection accuracy, we can detect an ongoing action when we only observe 42% of the action. Meanwhile, if we do not know the start of the action, we will have to wait until we observe 62% of the action.

There are some other interesting facts from Fig. 2. First, for all methods, the classification accuracy generally increases as we observe more and more of the action. It decreases a little bit at the end of the action due to the winding down of the action. The SVM method uses segment-based features and therefore poorly if the segment is the entire observation sequence. For SVM, the best approach to handle unknown starting point is to use a sliding window. Sliding window, however, is not the best approach for LSTM. LSTM is a state-based method with a built-in mechanism to forget, memorize, and retrieve information from history. As such, evaluating the LSTM from the beginning works better than using a sliding window. The HMM method is another state-based model, but this model is trained on short and segmented sequences of human actions, unlike the case of the LSTM method. This explains why the HMM method does not work as well as the LSTM method. This also explains why evaluating the HMM from the beginning leads to very poor performance.



(a) SVM method (b) LSTM method (c) HMM method

Fig. 2. Early action recognition performance SVM, LSTM, HMM methods. In all figures, the horizontal axis indicates the observational ratio, which is the proportion of an action that has already occurred the time the classification decision is made. The vertical axis shows the accuracy of the classifier, averaged over all action and non-action subsequences in the long testing sequences of multiple actions. When the start of the ongoing action is not given, one can either: look all the way back to the beginning, consider a fixed length history, or consider multiple starting points (sliding window). There is a large performance gap between knowing and not knowing the starting point, indicating the need for an accurate estimate of the starting point.

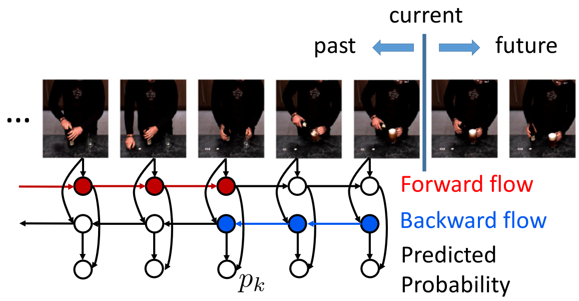
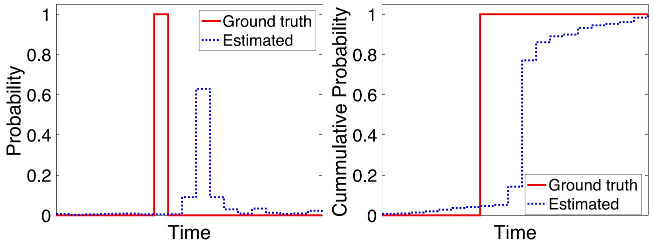


Fig. 3. Bidirectional LSTM for action starting point detection. At current time, we need to find the start point of ongoing action. Our network takes the input $\mathbf{s}_{1:T}$ of length T and outputs the probability of being the start point denoted as $p_{1:T}$. For a time k, the probability of it being the start point depends on both sequences before and after k. BLSTM provides a way to integrate these information flow.



(a) Probability density (b) Cumulative distribution

Fig. 4. Comparison between two loss functions. (a): two probability density functions for the location of the starting point. Directly taking the difference between these two curve leads to a loss value that is insensitive to the amount of mistake. (b) corresponding cumulative density functions for the probability functions in (a). The difference between these two curve indicates the level of mismatch between the predicted and ground truth values.

4. Starting point detection

We propose here a simple and effective method for estimating the starting point of an ongoing action. Our method is based on a Bidirectional LSTM (BLSTM) network (Schuster and Paliwal, 1997; Graves and Fernández, 2005).

Suppose we are at current time step t and we would like to find the start time of the ongoing action. The start of the ongoing action must be a transition point between two actions or between a non-action sequence and an action sequence. Consider a time k before t (i.e., $k < t$), the probability for k to be a transition point can be estimated by comparing the sequence of human motion before k and the sequence of human motion after k. This is why we propose to use a Bidirectional LSTM instead of a unidirectional LSTM. Bidirectional LSTM can keep two separate information flows: forward and backward. At time k the forward information flow until k and the backward information flow from t back to k can be combined (and therefore compared and contrasted) to predict the probability that k is a transition point. Fig. 3 shows our network architecture

The input to our network is a sequence of human motion $\mathbf{s}_{1:T}$ of length T, and the output is a sequence of the same length $p_{1:T}$ with p_k indicating the probability for k to be the start time of the ongoing action. During training, we know the ground truth starting time, and we define a target output sequence $y_{1:T}$ with $y_k = 1$ if k is the starting point and $y_k = 0$ otherwise.

Our goal is to train a Bidirectional LSTM so that the predicted probability is the same as the ground truth value, i.e., $p_k = y_k$. One naive solution is to define the training loss using the sum of squared errors, i.e., $\sum_{k=1}^{T} \|p_k - y_k\|^2$. However, this loss function is not a suitable choice because it is insensitive to the amount of prediction error. The loss is the same no matter how far away the predicted value from the ground truth value. Similar to the sum-of-squared-errors loss, the cross entropy loss $-\sum_{k=1}^{T} y_k \log(p_k)$ also has little tolerance for prediction error and therefore is not suitable for our starting point prediction.

Viewing the predicted sequence $p_{1:T}$ and the target sequence $y_{1:T}$ as two probability density functions, we define the loss via the cumulative distribution functions instead:

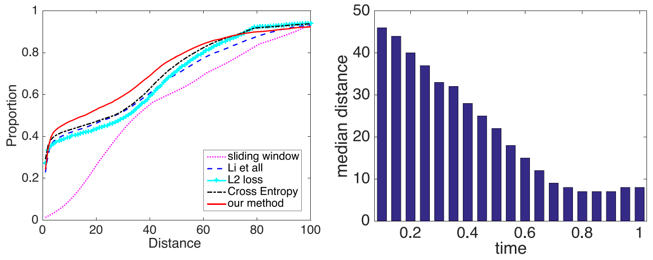
$$\mathcal{L}(\mathbf{s}_{1:T}) = \sum_{m=1}^{T} \left\| \sum_{k=1}^{m} p_k - \sum_{k=1}^{m} y_k \right\|_2^2. \quad (2)$$

Fig. 4 illustrates the benefits of defining the loss based on the difference of the two cumulative distribution functions.

It should be noted that $p_1, \ldots, p_T$ are normalized probability values. The direct output of the BLSTM network at each time k is unnormalized probability value $\bar{p}_k$, and we use the soft-max function to normalize them. That is:

$$p_k = \frac{\exp(\bar{p}_k)}{\sum_{j=1}^{T} \exp(\bar{p}_j)}. \quad (3)$$

We train a BLSTM to detect the start of an ongoing action using the above loss function. During the detection step, to estimate the start of the ongoing action, it is not necessary to consider the entire sequence of observations from the beginning. We propose to use a fixed-length look back window of size T, where T is big enough to cover most cases. In our experiments, T is set to the 95 percentile of the action lengths. At



(a) (b)

Fig. 5. Localization error analysis. (a): Cumulative distribution of the distance between predicted starting point and ground truth. A point (x, y) on a curve means: y is the proportion of the predicted starting points that are within x time steps of the ground truth values. Our method outperforms other approaches. (b) Distance between predicted starting point and ground truth starting point as a function of observation ratio. The localization error becomes smaller as the proportion of the ongoing action increases.

a time step t, we feed the observation sequence from $t - T$ to t to the BLSTM network, and the frame with the highest predicted probability is taken as the estimated starting point of the ongoing action.

Because the BLSTM network only needs to output prediction results for testing sequences of length T, we only train the network with training sequences of length T. Specifically, we sample multiple subsequences of length T from the original long action sequences. We discard the subsequences that do not contain any transition point. If there are multiple transition points in the sequence, only the last one is used as the start point of the ongoing action.

5. Experiments

In this section, we evaluate the performance of the proposed method for detecting the starting point of an ongoing action. Subsequently, we study the benefits of using the estimated starting point for early recognition.

5.1. Datasets

We perform experiments on three datasets: the Montalbano Gesture dataset (Escalera et al., 2014), MPII Cooking 2 dataset (Rohrbach et al., 2015), and ActivityNet dataset (Fabian Caba Heilbron et al., 2015). The former dataset has been described in Section 3.1. We now describe the latter two datasets.

The MPII Cooking 2 dataset. consists of 273 video sequences that vary in length from 40 s to 40 min, with a total of 2.8 million frames. The dataset contains 67 action classes and the number of examples for different actions varies drastically. For the purpose of our study, we sample top 10 action classes with the most number of examples. The action classes we use are: change temperature, close, dry, pour, put in, screw open, squeeze, stir, take lid, throw in garbage. We treat actions from all other classes as non-action.

For the MPII Cooking 2 dataset, we extract frames from each video at 15 frames per second and resize all frames to 224 × 224 pixels. We use the pre-trained two-stream network (Simonyan and Zisserman, 2014) to extract both spatial and temporal features from video. We divide each video into video segments of length 10. For each segment, we use the central frame to extract spatial features and the 10 optical flow images to extract temporal features. We use the output of the fc7 layers as spatial feature and temporal features, both yield a feature vector with 4096 dimensions. We concatenate both spatial and temporal feature vectors to obtain a representation vector for the video segment. The label for the segment is the label of the central frame. Due to computational cost, our experiments are performed on such down-sampled segments.

The ActivityNet dataset. (Release 1.3) comprises 20K videos of 200 activity categories collected from YouTube. The dataset is challenging due to uncontrolled environments, viewpoint and background variance within the same activity category. The lengths of the videos range from several minutes to half an hour. A single video may contain multiple activities and often also contains periods with none of the annotated activities. On average, 1.41 activities are annotated per video. The authors of ActivityNet did not release annotations for test set and the provided evaluation server only supports offline action detection evaluation metrics (mean averaged precision at different IOU threshold), which does not support online action detection and starting point detection. Thus, we use the validation set as our test set, and we use one fifth of the training set for validation.

We use the Temporal Segment Networks (Wang et al., 2016) that were trained on this dataset to extract both spatial and temporal features from video. Due to computational cost, we reduce the frame rate of the video sequences by a factor of 5 as follows. Each video is divided into segments of length 5. For each segment, we use the central frame to extract spatial features and the five optical flow images to extract temporal features. The feature vector for a video segment is the concatenation of the output of fc7 layers in both networks, with dimension 2048 and 1024 respectively. The class label for the segment is the label of the central frame.

5.2. Implementation details

The training data for the BLSTM for detecting the starting point of the actions should be video sequences of a fixed length T. For Montalbano Gesture dataset, we choose T to be 80, which is the 95 percentile of the action lengths. From the long training sequences of human actions, we sample training subsequences of length 80 and discard the subsequences that do not contain any action starting point. In total, we sampled 20,000 sequences for training and 9000 sequences for testing. We adopt a 2-layer Bidirectional LSTM with the memory size of 100. For MPII Cooking 2 dataset, we choose T to be 25 following the same criterion as Montalbano Gesture dataset. In total, we sampled 25,000 sequences for training and 9000 sequences for testing. We adopt a 1-layer Bidirectional LSTM with memory size of 100. For ActivityNet dataset, we choose T to be 150 and sampled 25,000 sequences for training and 8000 sequences for testing. We adopt a 1-layer Bidirectional LSTM with memory size of 200.

5.3. Localization errors for detecting the starting points

We use the absolute distance between the predicted starting point and the ground truth starting point as the performance measure. Fig. 5a shows the performance curves of our method and four other methods. The proposed method has a small error margin: 44.2% of the prediction errors are within 5 frames. The proposed method outperforms the popular sliding window approach, which determines the starting point based on the segment that yields the highest classification score. We also compare with a recently proposed method (Li et al., 2016) that jointly predicts the classification score and the starting point confidence value. This method uses a unidirectional LSTM and a Gaussian function to smooth the 0–1 loss between the predicted starting point and the ground truth value. This method does not work as well as ours (c.f., the median errors of 26 and 18). As can also be seen, the L_2 loss and the cross-entropy loss do not work as well as the proposed loss function.

We also study how the absolute distance between predicted starting point and actual starting point changes as we observe more and more of the ongoing action. Fig. 5b illustrates that as the observational ratio increases, the localization error becomes smaller. When the observational ratio exceeds 0.7, the localization error is impressively small, less than 7 frames. When the observational ratio is small (e.g., < 0.2), the localization error is large, which seems to be problematic. However, as shown in Fig. 2(b), when the observation ratio is small, the advantage of knowing start point over not knowing starting point is not obvious, because it is still very ambiguous to make recognition decision no matter how precise we can estimate the starting point.

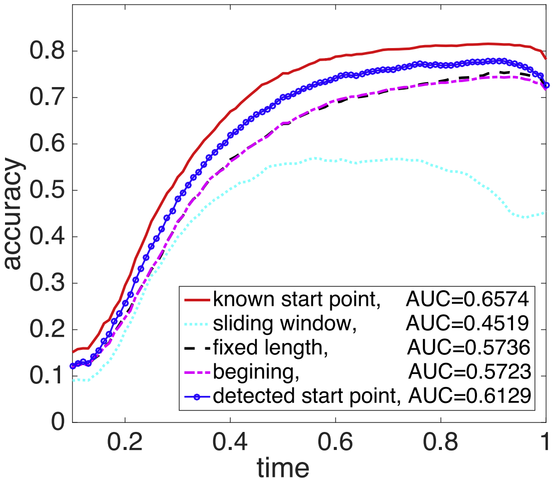


Fig. 6. Early recognition performance on the Montalbano Gesture dataset. If the starting point of the ongoing action is not given, the best approach is to use the proposed BLSTM network to estimate the starting point of the ongoing action. AUC denotes the area under the curve, a higher value corresponds to a better performance.

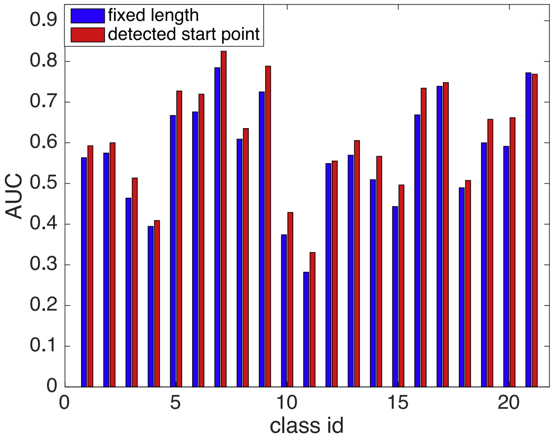


Fig. 7. Detailed AUC comparison for every class. The proposed approach outperforms the fixed-length approach for most cases.

5.4. Detecting starting points for early recognition

In this experiment, we use the predicted starting points for early recognition of ongoing actions. For classification, we choose to use the LSTM method described in Section 3 as it performs the best of all three methods in the experiments. Thus, we have two LSTM networks, one for action starting point detection and one for classification. Those networks are trained separately. During testing, first we use the action starting point detection network to predict the starting point for the sequence which ends at current time step. Then we evaluate the truncated sequence, from the predicted starting point to current time step, using the classification network.

Fig. 6 is an updated version of Fig. 2(b), adding the performance curve for the method that uses the predicted starting point. As can be seen, among all approaches for handling unknown starting point of the ongoing action, the approach that uses the predicted starting point achieves the best early recognition performance.

Fig. 7 shows the detailed comparison between the proposed approach and the *fixed length* approach. The fixed length approach is the second best approach, but it is outperformed by the proposed approach for almost all cases.

Fig. 8 compares several methods for early event recognition. Following Fawcett and Provost (1999); Nguyen et al. (2009), we use the Activity Monitoring Operating Characteristic (AMOC) curve (Fawcett and Provost, 1999) to evaluate the timeliness of detection. An AMOC curve shows the relationship between False Positive Rate and Normalized Time To Detection (NTtoD). To compute an AMOC curve, we vary the detection threshold and plot the curve of NTtoD versus FPR. Fig. 8 shows the AMOC curves for several methods: Structured-Output SVM (Tsochantaridis et al., 2005), MMED (Hoai and De la Torre, 2014), and LSTM with detected starting point. Because SOSVM and MMED are designed for binary detection, we also adapt the proposed method (LSTM) for binary detection. For each action class, we consider the binary detection task and there is a set of corresponding AMOC curves. Fig. 8 shows the AMOC curve on a representative class. Using LSTM with the predicted starting point can detect the action events faster than SOSVM and MMED at all false positive rates.

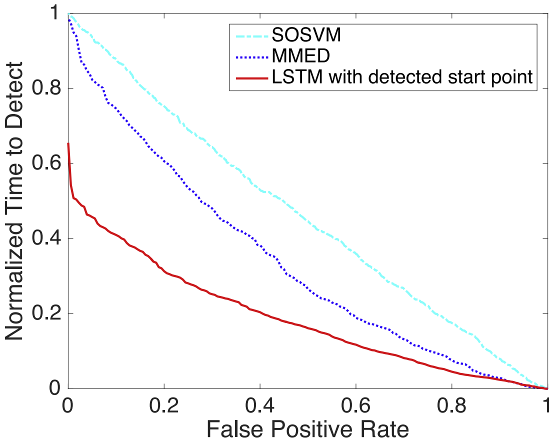


Fig. 8. Comparison of several methods for early event recognition. These figures show the AMOC curves for binary detection task. MMED (Hoai and De la Torre, 2014) is a method that is proposed for early event detection. Although it works better than SOSVM, it performs worse than LSTM that uses the predicted starting point.

Fig. 9 plots the recognition performance curves on the MPII Cooking 2 dataset. We use LSTM as our recognition method. In addition to showing the performance curve for the method that combines starting point detection with early recognition networks, the figure shows the performance curves for two other methods: (1) knowing the ground truth starting point, and (2) classifying the sequence which starts from beginning to current time step.

Fig. 10 shows the early recognition performance curve for the following methods: (1) combines starting point detection with our early recognition LSTM, (2) knowing the ground truth starting point, (3) classify the sequence which starts from beginning to current time step, (4) the approach of Ma et al. (2016) that using rank loss for early recognition. The proposed method, combining starting point detection and the LSTM for early recognition, outperforms LSTM with rank loss, a method specifically designed for early recognition. Note that, early recognition performance on the ActivityNet dataset is also reported in (Ma et al., 2016). However, that paper uses short evaluation sequences that contain a single action of interest. This evaluation situation does not correspond to the general problem of early recognition of ongoing actions. It is different from ours, so the results are not comparable.

5.5. Localization error versus accuracy

Our algorithm for estimating the starting point of an ongoing action is imperfect, and this affects the performance of the early recognition system. To understand how the localization error correlates with the decrease in accuracy, we perform the following experiment. During testing, we assume we have a method that can localize the starting point within a specific margin of error. For a fixed margin of error d, the method would return a random starting point from within $\pm d$ frames

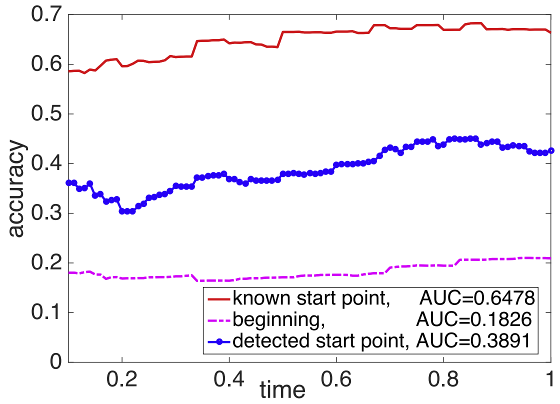


Fig. 9. Early recognition performance on the MPII Cooking 2 dataset. All methods use the LSTM recognition network. The differences are whether the starting point of an ongoing action is known and what to do when the starting point is not given.

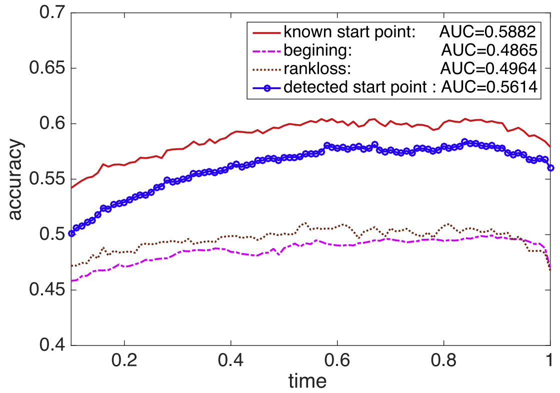


Fig. 10. Early recognition performance on the ActivityNet dataset. The figure shows early recognition performance using following method: (1) combines starting point detection with our early recognition LSTM, (2) knowing the ground truth starting point, (3) classify the sequence which starts from beginning to current time step, (4) early recognition with rank loss.

Table 1
Area Under the Curve (AUC) for several margins of localization errors. A localization method with an error margin of 5 means the estimated starting point is within 5 frames of the ground truth starting point. The proposed method that uses BLSTM to predict the starting point corresponds to having an error margin from 5 to 10 frames.

Margin of localization error	Recognition AUC
0 (known starting point)	0.6574
5	0.6307
10	0.5659
20	0.4911
Using the estimated starting point	0.6129

from the ground truth starting point (uniformly random). Table 1 shows the recognition performances at several margins of localization error on Montalbano Gesture dataset. Here the performance is measured as the area under the curve (AUC). As can be seen, using the proposed method to estimate the starting point corresponds to a margin error from 5 to 10 frames.

6. Conclusions

Most of the existing works for early recognition either make unrealistic assumptions about the detection process, they assume the start of an action is known, or they assume the computational models for partial actions can be used to localize the starting point. In this paper, first we study the importance of knowing the starting point of an ongoing action for early recognition. Through a set of experiments, we prove the necessity to estimate action starting point. Second, we propose a method for estimating the starting point. Our method is based on a Bidirectional LSTM network that integrates the information flow from both forward and backward directions to compute the probability for a point to be the starting point. Experiments on three human action datasets show that our starting point detection method improves early recognition performance.

Acknowledgments

This project is partially supported by the National Science Foundation, United States Award IIS-1566248 and Brookhaven National Lab, United States.

References

Basseville, M., Nikiforov, I.V., et al., 1993. Detection of Abrupt Changes: Theory and Application, volume 104. Prentice Hall Englewood Cliffs.

Berkes, I., Gombay, E., Horváth, L., Kokoszka, P., 2004. Sequential change-point detection in garch (p, q) models. 20, 1140–1167.

Cao, Y., Barrett, D., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., Lin, Y., Dickinson, S., Siskind, J.M., Wang, S., 2013. Recognize human activities from partially observed videos. In: Proc. CVPR.

Dave, A., Russakovsky, O., Ramanan, D., 2017. Predictive-corrective networks for action detection. arXiv preprint arXiv:1704.03615.

Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. In: Proc. CVPR.

Ellis, C., Masood, S.Z., Tappen, M.F., Laviol Jr, J.J., Sukthankar, R., 2013. Exploring the trade-off between accuracy and observational latency in action recognition. Int. J. Comput. Vis. 420–436.

Enikeeva, F., Harchaoui, Z., 2013. High-dimensional change-point detection with sparse alternatives. ArXiv:1312.1900.

Escalera, S., Baró, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce-López, H.J., Shotton, J., Guyon, I., 2014. Chalearn looking at people challenge 2014: Dataset and results. In: Proc. ECCV Workshops.

Fabian Caba Heilbron, Victor Escorcia, B.G., Niebles, J.C., 2015. Activitynet: A large-scale video benchmark for human activity understanding. In: Proc. CVPR.

Fawcett, T., Provost, F., 1999. Activity monitoring: Noticing interesting changes in behavior. In: Proc. KDD.

Graves, A., Fernández, J., 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks.

Hoai, M., De la Torre, F., 2014. Max-margin early event detectors, 107, 191–202.

Hoai, M., Lan, Z.Z., De la Torre, F., 2011. Joint segmentation and classification of human actions in video. In: Proc. CVPR.

Hoai, M., De la Torre, F., 2012. Maximum margin temporal clustering. In: Proceedings of International Conference on Artificial Intelligence and Statistics.

Hoai, M., Torresani, L., De la Torre, F., Rother, C., 2014. Learning discriminative localization from weakly labeled data. Pattern Recognit. 47, 1523–1534.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Hu, J.F., Zheng, W.S., Ma, L., Wang, G., Lai, J., 2016. Real-time rgb-d activity prediction by soft regression. In: Proc. ECCV.

Huang, D., Yao, S., Wang, Y., De La Torre, F., 2014. Sequential max-margin event detectors. In: Proc. ECCV.

Kiran, B.R., Thomas, D.M., Parakkal, R., 2018. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. J. Imaging 4, 36.

Kitani, K., Ziebart, B., Bagnell, J., Hebert, M., 2012. Activity forecasting. In: Proc. ECCV.

Kong, Y., Fu, Y., 2015. Max-margin action prediction machine. 38, 1844–1858.

Kong, Y., Kit, D., Fu, Y., 2014. A discriminative model with multiple temporal scales for action prediction. In: Proc. ECCV.

Lan, T., Chen, T.C., Savarese, S., 2014. A hierarchical representation for future action prediction. In: Proc. ECCV.

Lan, T., Wang, Y., Mori, G., 2011. Discriminative figure-centric models for joint action localization and recognition. In: Proc. ICCV.

Lee, H., Battle, A., Raina, R., Ng, A.Y., 2006. Efficient sparse coding algorithms. In: NIPS.

Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., Liu, J., 2016. Online human action detection using joint classification-regression recurrent neural networks. In: Proc. ECCV.

Luo, J., Wang, W., Qi, H., 2013. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In: Proc. ICCV.

Ma, S., Sigal, L., Sclaroff, S., 2016. Learning activity progression in lstms for activity detection and early detection. In: Proc. CVPR.

Malhotra, P., Vig, L., Shroff, G., Agarwal, P., 2015. Long short term memory networks for anomaly detection in time series. In: Proceedings, Presses universitaires de Louvain. p. 89.

Marchi, E., Vesperini, F., Eyben, F., Squartini, S., Schuller, B., 2015. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks. In: ICASSP.

Nguyen, M.H., Torresani, L., De l. Torre, F., Rother, C., 2009. Weakly supervised discriminative localization and classification: a joint learning process. In: Proc. ICCV.

Ni, B., Paramathayalan, V.R., Moulin, P., 2014. Multiple granularity analysis for fine-grained action detection. In: Proc. CVPR.

Oliver, N., Garg, A., Horvitz, E., 2004. Layered representations for learning and inferring office activity from multiple sensory channels. Comput. Vis. Image Underst. 96, 163–180.

Picard, D., 1985. Testing and estimating change-points in time series. Adv. Appl. Probab. 17, 841–867.

Poor, H.V., Hadjiliadis, O., 2009. Quickest Detection. volume 40. Cambridge University Press Cambridge.

Rabiner, L.R., 1989. A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE 77, 257–286.

Raptis, M., Sigal, L., 2013. Poselet key-framing: A model for human activity recognition. In: Proc. CVPR.

Rohrbach, M., Rohrbach, A., Regneri, M., Amin, S., Andriluka, M., Pinkal, M., Schiele, B., 2015. Recognizing fine-grained and composite activities using hand-centric features and script data. In: Proc. ICCV.

Rumelhart, D., Hinton, G., Williams, R., 1986. Learning internal representations by error propagation. In: Parallel Distributed Processing, vol. 1, MIT Press, Cambridge, MA, pp. 318–362, chapter 8.

Ryoo, M., 2011. Human activity prediction: Early recognition of ongoing activities from streaming videos In: Proc. ICCV.

Ryoo, M., Aggarwal, J.K., 2009. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: Proc. ICCV.

Ryoo, M.S., Fuchs, T.J., Xia, L., Aggarwal, J.K., Matthies, L., 2015. Robot-centric activity prediction from first-person videos: What will they do to me?. In: International Conference on Human-Robot Interaction.

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45, 2673–2681.

Shou, Z., Pan, J., Chan, J., Miyazawa, K., Mansour, H., Vetro, A., Giro-i Nieto, X., Chang, S.-F., 2018. Online detection of action start in untrimmed, streaming videos. The European Conference on Computer Vision (ECCV).

Simonyan, K., Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. In: NIPS.

Singh, G., Saha, S., Cuzzolin, F., 2017. Online real time multiple spatiotemporal action localisation and prediction. In: Proc. ICCV.

Soomro, K., Idrees, H., Shah, M., 2016. Online localization and prediction of actions and interactions. ArXiv:1612.01194.

Tian, Y., Sukthankar, R., Shah, M., 2013. Spatiotemporal deformable part models for action detection. In: Proc. CVPR.

Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., 2005. Large margin methods for structured and interdependent output variables. J. Mach. Learn. Res. 6, 1453–1484.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York, NY.

Veeriah, V., Zhuang, N., Qi, G.J., 2015. Differential recurrent neural networks for action recognition. In: Proc. ICCV.

Vondrick, C., Pirsiavash, H., Torralba, A., 2016. Anticipating the future by watching unlabeled video. In: Proc. CVPR.

Wang, Y., Hoai, M., 2016. Improving human action recognition by non-action classification.

Wang, B., Hoai, M., 2018a Predicting body movement and recognizing actions: an integrated framework for mutual benefits.

Wang, Y., Hoai, M., 2018b. Pulling actions out of context: Explicit separation for effective combination.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Va. Gool, L., 2016. Temporal segment networks: towards good practices for deep action recognition. In: Proc. ECCV.

Wei, Z., Wang, B., Hoai, M., Zhang, J., Lin, Z., Shen, X., Mech, R., Samaras, D., 2018. Sequence-to-segment networks for segment detection.

Weston, J., Chopra, S., Bordes, A., 2014. Memory networks. ArXiv:1410.3916.

Xia, L., Chen, C.C., Aggarwal, J., 2012. View invariant human action recognition using histograms of 3d joints. In: Proc. CVPR Workshops.

Xie, Y., Huang, J., Willett, R., 2013. Change-point detection for high-dimensional time series with missing data. IEEE J. Sel. Top. Sign. Proces. 7, 12–27.

Xu, Z., Qing, L., Miao, J., 2015. Activity auto-completion: Predicting human activities from partial videos. In: Proc. ICCV.

Yacoob, Y., Black, M.J., 1999. Parameterized modeling and recognition of activities. Comput. Vis. Image Underst. 73, 232–247.

Yang, J., Yu, K., Gong, Y., Huang, T., 2009. Linear spatial pyramid matching using sparse coding for image classification. In: Proc. CVPR.

Yu, G., Liu, Z., Yuan, J., 2015. Discriminative orderlet mining for real-time recognition of human-object interaction. In: Proc. ACCV.

Yu, G., Yuan, J., 2015. Fast action proposals for human action detection and search. In: Proc. CVPR.

Yuan, J., Liu, Z., Wu, Y., 2009. Discriminative subvolume search for efficient action detection. In: Proc. CVPR.

Yue-He. Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G., 2015. Beyond short snippets: Deep networks for video classification. In: Proc. CVPR.

Zanfir, M., Leordeanu, M., Sminchisescu, C., 2013. The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection. In: Proc. ICCV.