Temporal Action Localization with Pyramid of Score Distribution Features

Jun Yuan[1], Bingbing Ni[2], Xiaokang Yang[2], Ashraf A.Kassim[1]
[1]National University of Singapore, [2]Shanghai Jiao Tong University
yuanjun@nus.edu.sg, nibingbing@sjtu.edu.cn, xkyang@sjtu.edu.cn, ashraf@nus.edu.sg

Abstract

We investigate the feature design and classification architectures in temporal action localization. This application focuses on detecting and labeling actions in untrimmed videos, which brings more challenge than classifying presegmented videos. The major difficulty for action localization is the uncertainty of action occurrence and utilization of information from different scales. Two innovations are proposed to address this issue. First, we propose a Pyramid of Score Distribution Feature (PSDF) to capture the motion information at multiple resolutions centered at each detection window. This novel feature mitigates the influence of unknown action position and duration, and shows significant performance gain over previous detection approaches. Second, inter-frame consistency is further explored by incorporating PSDF into the state-of-the-art Recurrent Neural Networks, which gives additional performance gain in detecting actions in temporally untrimmed videos. We tested our action localization framework on the THUMOS'15 and MPII Cooking Activities Dataset, both of which show a large performance improvement over previous attempts.

1. Introduction

Action recognition is an active research area with many potential applications in multimedia, home care, logistic support and security-based applications. With the technology advancement of network infrastructure, social media and video-capturing devices, the explosion of video contents has greatly lifted the demand of automatic content analysis. There are considerable amounts of work on analyzing human activities in videos; many pieces of work focus on action classification, labeling human actions with various datasets [30, 16, 34, 39, 28, 13]. It is not a easy problem since human activities often vary in pose, interaction, clothing and background. During the last decade, great progress has been made on both the dataset quality and recognition algorithms.

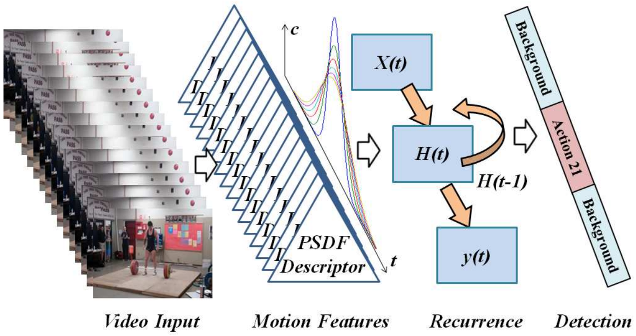

Figure 1. Our framework for temporal action localization. We encode visual features at multiple temproal resolutions, combine with class SVM scores to form our Pyramid of Score Distribution Features (PSDF). The features are further processed by Recurrent Neural Networks (RNN) to produce action detection results.

Despite the great improvements in action classification, there are still drawbacks in this field. Most human action classification algorithms are developed on datasets of trimmed videos [26, 30, 16, 20], where each video file is aligned with a single human action. The videos are often trimmed by hand. This setting simplifies the real-life video content analysis, however it is most of the time unrealistic, since a real video often contains multiple action instances as well as irrelevant backgrounds, i.e. the video is untrimmed. A more practical setting is to label every video frame with action classes; this can be considered as a detection task which brings in new challenges due to the uncertainty in action occurrence, where the action label, position and duration are all unknown.

This per-frame labeling process utilizes both the current frame action information and inter-frame consistency. It can be considered as an automatic trimming processes, including the detection of action start / stop time and classifying the underlying action at the same time. This task is much more demanding than outputting a label from a video file, and requires huge amount of temporally labeled video data. Due to the cost of annotation, the studies in the temporal action localization problem are still developing at a primitive stage [25], with a limited number of temporally annotated datasets.

The THUMOS Action Recognition Challenge [9] is developed to address this issue. The dataset provides a large amount of untrimmed videos from real-life human activities with temporal annotations, with which it is viable to further investigate temporal action localization under the realistic setting. Yet, the problem is still difficult in practice. First, the scale of sliding window is hard to choose. Since the true duration can vary vastly, a single size detection window can hardly match the effective portion of the underlying action. Second, the multi-resolution approach to address the scaling issue often results in high dimensionality if the multi-scale features are simply concatenated; if the features are scored, a simple non-maximal suppression used in [10, 25] is prone to noise and might not well exploit the underlying feature distribution.

To address those difficulties, we propose the Pyramid of Score Distribution Features (PSDF) for the temporal action localization task. This feature aims at capturing the multi-resolution context around anchor frames. Each anchor frame serves as the center of context pyramid, and each scale of the pyramid is described with action scores extracted with Fisher Encoded Improved Dense Trajectories (IDT) [35]. This multi-resolution feature is straight forward to understand: if one scale is close to the correct action occurrence, its corresponding score should be high, together with its neighboring scales. In other words, we utilize the multi-resolution action response distribution for each anchored detection window and explore the action information from various temporal scales centered on this anchor point. This multi-resolution approach is more robust than a single scale feature representation where only the information from a fixed length window is considered. By exploiting the distribution among these scores, we can use a classifier to label the frames. We test our features on the THUMOS'15 [9] Dataset and MPII Cooking Activities Dataset [25].

To further explore the temporal consistency, we combine our PSDF descriptor with the state-of-the-art Recurrent Neural Networks (RNN). RNN is widely used in sequential data analysis, with a natural Markovian structure to utilize context information. The PSDF focuses on confidence level at current, while the combination with RNN gives a better modeling of its temporal transition - both consistency over one action and switching between different actions. Sometimes the windows at current frame appear to be noisy, nonetheless the representation can be much more robust with the propagation of previous multi-resolution contexts. We use two types of RNN, the Elman-Net [7] and Long Short Term Memory (LSTM) [12] networks on our PSDF descriptor for detection tasks; these non-linear classifiers have higher precision over linear ones, with utilization of temporal feedbacks. The framework is shown in Figure 1.

The rest of this paper is organized as follows: Section 2 briefly introduces the related work on human action recognition. Section 3 describes our Pyramid of Score Distribution Features (PSDF) for temporal action localization. The further refinements on action detection with Elman-Net and LSTM networks are covered in Section 4. The experiment results are reported in Section 5. Section 6 proposes potential improvements to our framework as future studies. Section 7 concludes this paper.

2. Related Work

A substantial amount of work has been developed in the area of action recognition. There are various human action datasets built for algorithm development, on sources from movies, surveillance videos as well as daily captures [25]. An extensive list can be found in [2]. The KTH, UCF-101, HMDB-51 and Hollywood2 [26, 30, 16, 20] datasets are frequently used in human activity classification. The Coffee and Cigarettes, High Five, MSR Action Dataset and MEXaction2 [18, 21, 41, 31] are datasets with action detection tasks. Some datasets, like the MPII Cooking Activities Dataset [25], are developed for both classification and detection tasks.

Many datasets for temporal action localization are small, and with highly unbalanced action classes. This is due to the high cost of temporal annotation, and often results in overfitting in detection algorithms. Thus the studies of temporal localization are still quite primitive. The THUMOS Action Recognition Challenge [9], first held in 2013, is a significant attempt to bring in large scale dataset for both classification and detection; it has become an important platform for exploring new approaches in action studies. The THUMOS'15 dataset contains over 430 hours of video data and 45 million frames [9] for untrimmed action classification and temporal localization, based on which we develop our localization framework.

The methodologies for action recognition have been developing fast. Earlier attempts include space-time features like STIP [17] and extensions on classical image descriptors [27, 37, 14, 40]. Heng Wang and Cordelia Schmid [34] proposed the Dense Trajectories (DT), which is widely used in subsequent action recognition studies. This feature tracks points on dense optical flow field, and encodes the trajectories with local descriptors. They further improved their DT features (IDT) by offsetting camera motion, and use Fisher Encoding to improve the classification performance [35]. Minh Hoai et al. [11] proposed a score ranking method for action classification. They partitioned the video at global level with different granularity, formed Fisher Vectors [22] (FV) and ranked the action scores to an ordered distribution

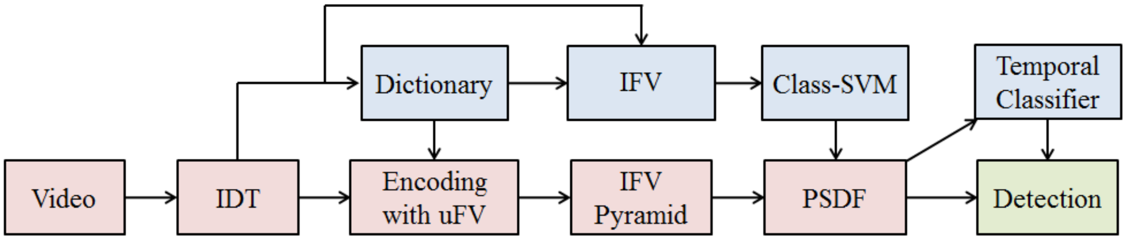

Figure 2. Our temporal localization framework. The boxes in the first row are generated with features from training set.

for classification. Our PSDF is partially inspired by their work, while we use score distribution locally rather than globally; it is extracted at anchor frames at multiple scales to form a robust descriptor for action localization.

For detection tasks, the High Five [21] and Coffee and Cigarettes [18] datasets are earlier attempts for action localization. Both focus on spatial-temporal detection which is more challenging, however these datasets are small and the methodologies have yet to be tested on large-scale datasets. Yuan et al. [41] proposed an efficient pattern searching algorithm. Marcus et al. [25] proposed the MPII Cooking Activities Dataset which contains fine-grained human activities; the dataset contains both classification and temporal localization tasks. The detection is based on DT features, with integrated histogram for efficient computation. Stoian A. et al. [31] developed the MEXaction2 dataset for two-class action retrieval; their baseline methodology is quite similar to [35], with Fisher Encoding on DT features. In THUMOS'14, the CUHK submission [36] applied both IDT and Convolutional Neural Networks (CNN) features with a single resolution sliding window. The FV from motions and average pooled CNN features from scenes are fused to give the detection labels on video segments. The INRIA submission [10] used IDT features only, but applied sliding windows at multiple resolutions; its methodology is similar to [25], where a simple non-maximal suppression is applied to separate detection windows. They also used many post-processing techniques to refine detection labels. In our work, we take a more structural approach by using classifiers on distributions, which better utilizes context information and generates more robust results.

The neural networks has achieved great success in computer vision applications. The CNN [19] has been dominating in image classification [15, 29], and its intermediate features generated by hidden layers have been widely used in object detection, segmentation and saliency applications. Karen et al. [28] proposed a two-stream CNN for action recognition in videos. Andrej et al. [13] proposed various CNN fusion models for large-scale video classification. Xu et al. [39] proposed the latent concept descriptors (LCD), combined with Fisher / VLAD encoding on different layers of CNN intermediates, which gives comparative performance with IDT features. This method is used by most participants in the classification task of THUMOS'15. The RNN has also been widely used in video content analysis. Pedro et al. [24] proposed a recurrent structure for scene labeling. The LSTM structure was incorporated by Jeff D. et al. [6] and Wonmin B. et al. [5] respectively in video recognition and scene labeling. Wu. et al. [38] proposed a hybrid deep network for video classification, which uses LSTM networks on top of spatial and temporal CNN features. Vivek V. et al. [33] proposed a differential RNN for action recognition, emphasizing the information gain with the salient motions between successive frames. These works have more focus on classification, while our paper emphasizes more on temporal action localization.

3. Pyramid of Score Distribution Features

3.1. Motivation

In a typical action localization task, the dataset normally consists of untrimmed videos with multiple actions mixed with background movements. Neither the action label, position nor the duration is known and a per-frame labeling process is generally required. This is different from the trimmed action classification tasks, where only one action is contained in a file and a video-level pooling is applicable.

Deciding the sliding window size can be challenging in the localization task. A window too small can only cover a small action fragment, makes the action features noisy and causes extra computation cost; while a window too large will include lots of background movements and make the feature biased from its real distribution. The multi-resolution approaches are used in [25, 10] to address the problem; they use action scores instead of raw features to avoid high dimensionality, and use the max scores to separate detection windows. Each window is classified with one action for the localization task. However, using noisy max scores is not robust enough and fails to utilize more abundant distribution information, which motivates our structural approach.

In our experiments, we find the action response are cor-

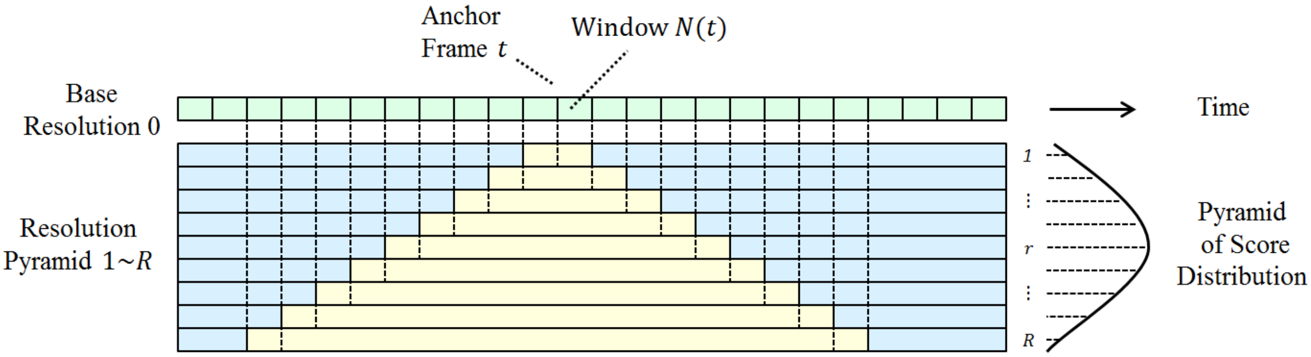

Figure 3. The formation of PSDF descriptor at anchor frame t. We use unnormalized Fisher Vector (uFV) at the base resolution and compute the uFV recursively in all resolution levels. The additive uFV is re-normalized, combined with class SVM to generate score distributions. The score should be high at the closest window coresponding to the underlying action. The PSDF is a straight forward descriptor, since the score distribution contains information of action label, position and duration.

related in different temporal scales. If an action is found at one scale, its neighboring scales are still representative for this instance. Specifically, the action scores should satisfy a distribution in a temporal pyramid. Based on this observation, we propose a novel Pyramid of Score Distribution Features (PSDF). A fixed-size sliding window is applied across the untrimmed video at evenly distributed anchor frames to detect potential action positions. The process is repeated at multiple temporal resolutions to cover different action durations. Each window is treated as an video segment and the action classification pipeline in [35] is applied. We use the class SVM scores instead of encoded trajectory features as descriptor at the anchor frames. The PSDF is later used to train temporal action detectors, with SVM or RNN. The process is illustrated in Figure 2.

The PSDF is an intuitive feature descriptor: it represents the confidence level of action class, position and duration. For example, if an action is found in resolution r, the confidence should be high at this correct temporal scale, together with its neighbors, as in Figure 3. We can train a linear SVM on the distribution from each anchor frame to detect the labels. Since this SVM is directly related to temporal localization, we call it Temporal-SVM to avoid confusion with the Class-SVM used to generate action scores. Alternatively, we can use the PSDF to train more complicated RNN classifiers.

The benefits of our approach lies in the following:

1. The use of multi-resolution naturally captures the information spanned in temporal scales. A single scale [36] might not contain enough discriminative information, while a temporal pyramid is more comprehensive and provides significantly better performance.

2. The Temporal-SVM is a better classifier compared to the crude max score approach; the latter is more prone to noise and fails to fully utilize the distribution information. The PSDF also allows us to apply more complex classifiers as in Section 4.

3. The PSDF descriptor can be calculated in a linear recursive way which greatly accelerates the computation, as shown in the following section.

3.2. Feature Encoding in Multiple Resolutions

We score the IFV at each anchor frame and across all the temporal resolutions to form our PSDF descriptor. Due to the high cost of computation, we modify the IFV process so that the multi-scale representation can be obtained by a linear combination from a base resolution, as in Figure 3.

We use the IDT [35] with Improved Fisher Encoding to get the IFV [23] in a single sliding window. The IDT features are compressed by PCA, and clustered with Gaussian Mixture Models (GMM). The clusters are denoted as tuple $\{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mid k = 1, 2, \dots K\}$. Given all the IDT features in a window $F = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \boldsymbol{x}_N)$, we apply Fisher Encoding [22] on each of the sub-features:

$$u_{jk} = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} q_{ik} \frac{x_{ji} - \mu_{jk}}{\sigma_{jk}}, \tag{1}$$

$$v_{jk} = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} q_{ik} \left[\left(\frac{x_{ji} - \mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right]. \tag{2}$$

u_{jk} and v_{jk} stand for the deviation brought by all the samples at element j and mode k. q_{ik} stands for the posterior probability of sample i at mode k:

$$q_{ik} = \frac{\exp\left[-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right]}{\sum_{m=1}^{K} \exp\left[-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_m)\right]} \tag{3}$$

The FV is obtained by concatenating the vector $\boldsymbol{u}$ and $\boldsymbol{v}$. Further applying signed square-rooting and L_2 normalization on the FV elements yields the IFV. An Class-SVM can be trained to classify actions; this is the standard pipeline in a trajectory based action classification. In detection tasks, we use the scores generated by this Class-SVM (from the training set) as descriptor in a sliding window.

It is expensive to score the IFV on all the anchor frames across all the resolutions; for R resolutions and A anchor frames, this results in AR computations for IFV. Since the number of features increases with temporal scales, the overall complexity can be $\mathcal{O}(AR^2)$, which is extremely costly. To accelerate the computation, we unnormalize the FV (uFV) to make them additive by taking away the normalization factor $N\sqrt{\pi_k}$ and $N\sqrt{2\pi_k}$. We define a base resolution 0 sampled at half of resolution 1, and use the uFV computed at this level to get uFV for other resolutions. Let r stand for the resolution level, $N(t)$ stand for the set of features at base resolution 0 after anchor frame t, we have:

$$\phi_{t,0} \triangleq u'_{jk,t,0} = \sum_{i \in N(t)} q_{ik} \frac{x_{ji} - \mu_{jk}}{\sigma_{jk}}, \tag{4}$$

$$\psi_{t,0} \triangleq v'_{jk,t,0} = \sum_{i \in N(t)} q_{ik} \left[\left(\frac{x_{ji} - \mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right], \tag{5}$$

$$\phi_{t,1} = \phi_{t-1,0} + \phi_{t,0}, \tag{6}$$

$$\psi_{t,1} = \psi_{t-1,0} + \psi_{t,0}, \tag{7}$$

$$\phi_{t,r} = \phi_{t,r-1} + \phi_{t-r,0} + \phi_{t+r-1,0}, \tag{8}$$

$$\psi_{t,r} = \psi_{t,r-1} + \psi_{t-r,0} + \psi_{t+r-1,0}. \tag{9}$$

The above recursive addition only needs to compute the uFV once at the base resolution, which greatly reduces the computation cost. FV can be obtained by re-normalizing with the factor $\sqrt{\pi_k}$ and $\sqrt{2\pi_k}$; the factor N is ignored since it gets canceled out by subsequent IFV normalization. The IFV of all sliding windows can be computed efficiently by moving down the anchor frame. Since the computation is dominated by generating uFV, this approach reduces the complexity from $\mathcal{O}(AR^2)$ to $\mathcal{O}(A)$.

We can get the PSDF by applying the Class-SVM on the IFV at anchor frames. A Temporal-SVM can be trained on the PSDF to directly detect actions, or as the input of more complicated network classifiers.

4. Recurrent Neural Networks for Action Detection

While the PSDF describes the confidence of action class, position and duration at anchor frames, it is desired to explore the evolution of this feature with temporal structures. Within the same action instance, the PSDF at neighboring anchor frames should have consistency; while at transition points, the neighboring PSDF should have distinct distributions. Also, the feature becomes more robust to noise with temporal contexts. To model the temporal evolution of PSDF in action localization, we combine the feature with the state-of-the-art Recurrent Neural Networks (RNN). The multi-resolution context from PSDF propagates in the network, allowing a more comprehensive representation of temporal action information.

4.1. Recurrent Structures

We use two types of RNN, the Elman-Net [7] and LSTM [12] networks to generate detection labels at anchor frames. The RNN structure combines the current input with previous states, propagating the context information to subsequent time steps. A typical feedback structure of RNN is shown in Figure 1. We use the PSDF as the network input, and class labels as the network output.

Elman-Net. The Elman-Net linearly combines the current input and the hidden state from the last time step. The current hidden state is generated via a non-linear activation on the combination, typically a $sigmoid$ or $tanh$ function. A softmax classifier as the output layer is applied to recognize the underlying data. The forward propagation of Elman-Net is as follows:

$$\boldsymbol{h}_t = \tanh(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1} + \boldsymbol{b}_h), \tag{10}$$

$$\boldsymbol{y}_t = softmax(\boldsymbol{W} \boldsymbol{h}_t + \boldsymbol{b}). \tag{11}$$

LSTM. The LSTM networks are developed to address the vanishing / exploding gradients in the training process, by enforcing constant error flow in its memory cell [12]. The cell has of four elements: an input gate, an output gate, a forget gate and a self-recurrent connection with a weight of one to ensure constant error flow. The gates serve as nonlinear units, modulating the interaction between the cell and current input / output, as well as its previous states. We use the setting in [1] to determine the gate and cell candidate states:

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i), \tag{12}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f), \tag{13}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o), \tag{14}$$

$$\widetilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c). \tag{15}$$

The cell state and output are determined:

$$\boldsymbol{C}_t = \boldsymbol{i}_t * \widetilde{\boldsymbol{C}}_t + \boldsymbol{f}_t * \widetilde{\boldsymbol{C}}_{t-1}, \tag{16}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{C}_t). \tag{17}$$

$$\boldsymbol{y}_t = softmax(\boldsymbol{W} \boldsymbol{h}_t + \boldsymbol{b}). \tag{18}$$

The experiment setting in our detection task is similar to Elman-Net, with PSDF arranged by time steps as input and anchor frame labels as output. Since the PSDF is already a descriptive feature, we only use one hidden layer in these networks.

4.2. Smoothing Regularizer

The PSDF contains action scores from C classes and R scales, together with dimension RC. The scores from different resolutions are often correlated, e.g. they should not vary vastly at neighboring scales. As a result, the corresponding network weights applied to PDSF should also present a similar internal structure. To incorporate this characteristic, we apply a smoothing loss on the network weights:

$$J(\boldsymbol{W}_x) = \sum_{c=1}^{C} \sum_{r=1}^{R-1} \parallel w_{x(c,r+1)} - w_{x(c,r)} \parallel_2^2 \tag{19}$$

This regularizer results in about 1% performance improvement in our detection experiments. It is combined with the L_2 regularizer to limit the effective degree of freedom and promote better generalization capabilities.

4.3. Practical Considerations

Unbalanced Data. The background actions often dominate the untrimmed videos in the localization task, and the number of action instances from different classes can vary vastly. These result in a highly uneven composition of labels. To address the problem, we apply different weights, typically inverse proportional to the action instances in each class, in training the Temporal-SVM and RNN. The class balancing gives improved classifier performance in the detection task.

Different Video Lengths. The RNN does not require uniform sequence duration; nonetheless padding the sequences in a batch to the same length will allow fast parallel processing. Also, it is practical to break very long sequences into smaller fragments, since the correlation with long time lags is generally small. For the simplicity of training, we break the video sequences into shorter fixed-length segments, with the tail padded with zeros.

5. Experiments

We report the experiment results of our framework on THUMOS'15 [9] and MPII Cooking Activities Dataset [25], and compare with previous state-of-the-art attempts. Both datasets are large-scale and serve as benchmarks of many related studies. We also report the experiments with different resolution levels and classification methods to demonstrate the strength of our approach. The PSDF outperforms the benchmarks, while the combination with RNN further boosts the detection performance. Extensive evaluation is still difficult in action localization because of the lack of datasets and different evaluation standards.

5.1. The THUMOS'15 Challenge

The THUMOS Action Recognition Challenge is the largest action recognition challenge, benchmarking the performance in both action classification and detection. The training set of THUMOS'15 Challenge is the UCF-101 [30] dataset, containing over 13,000 trimmed videos from 101 classes. The validation, test and background sets contain approximately 2100, 5600 and 3000 untrimmed videos from YouTube. We resize the videos so that their long edges has a fixed length of 320 pixels.

In the temporal action localization task, only 20 out of the 101 classes need to be detected. The action classes as well as its temporal annotation need to be retrieved. The evaluation is based on Interpolated Average Precision (AP), and mean AP (mAP) averaged over all classes. A detection R_p is considered correct if its overlap o with ground truth R_{gt} is over a specified value:

$$o = \frac{R_p \cap R_{gt}}{R_p \cup R_{gt}} \tag{20}$$

The given overlap takes five discrete values, ranging from 0.1 to 0.5, as specified in the evaluation toolbox.

We use the training set to generate PCA projection matrix with a reduction factor of 2, and 256 GMM clusters. We use the openCV implementation on IDT extraction [35] and VLFeat [32] library on GMM clustering. The VLFeat library for Fisher Encoding is modified so that it can compute uFV efficiently; this significantly accelerates the computation of PSDF descriptors. The background set is combined with the training set to generate one-vs-the-rest Class-SVM under the liblinear [8] implementation. Thus for each given video segment, we can get 101 class scores based on IFV pooling on IDT features.

The validation set is used to generate the Temporal-SVM. To generate PSDF, we use nine resolutions from 10 to 90 frames with increment of 10 frames. The base resolution to generate uFV is chosen at 5 frames (∼0.2s), which is also the distance between anchor frames. The short period is chosen to ensure the instantaneous actions are not diluted by backgrounds. We use a full score distribution on 101 classes, so each anchor point has 101*9 = 909 action scores which are used for training the Temporal-SVM. For demonstration of the score distribution, we manually crop a video clip and show the scores from different resolution levels in Figure 4.

The detection results on the testing set of THUMOS'15 is given in Table 1 with PSDF and Temporal-SVM. Our results outperform the previous state-of-the-art submission on THUMOS'14, given that the size of testing set has significantly expanded from 1500 to 5600 videos. The THUMOS'15 validation set is constructed from THUMOS'14 validation and test set, so it is possible to evaluate our framework on this set for more comparative results. Although some of the non-detection videos (not belonging to the 20 detection classes) are not included, these background videos turn out to have very limited impact on the detection

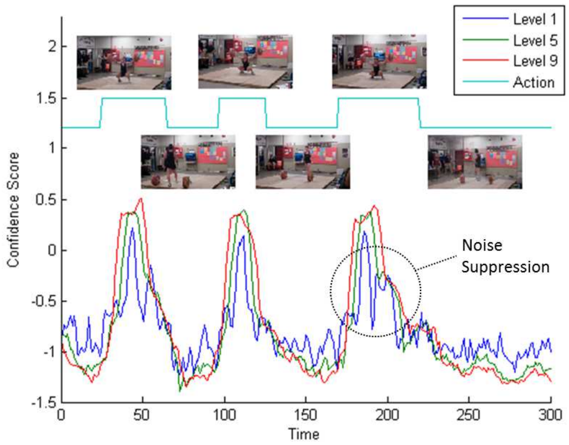

Figure 4. Confidence scores with action instances. The scores are higher at the corresponding position and duration, and smoother at larger temporal scales. Scores from different resolutions are highly correlated, making the representation more robust.

mAP		0.1	0.2	0.3	0.4	0.5
TM'14	CUHK [36]	18.2	17.0	14.0	11.7	8.3
	INRIA [10]	36.6	33.6	27.0	20.8	14.4
	Ours.	**51.4**	**42.6**	**33.6**	**26.1**	**18.8**
TM'15	Ours.	**40.9**	**36.3**	**30.8**	**23.5**	**18.3**

Table 1. Detection results on THUMOS'14 and THUMOS'15 Datasets.

mAP	0.1	0.2	0.3	0.4	0.5
Max Classifier	45.6	36.8	28.6	22.1	16.7
T-SVM (Ours.)	**51.4**	**42.6**	**33.6**	**26.1**	**18.8**

Table 2. Max Vs. SVM classifier on PSDF descriptor. Temporal-SVM utilizes distribution information and gives better performance.

performance.

We also provide experimental results with different number of resolution levels and use of max classifier (used in non-maximal suppression) in Figure 5 and Table 2. The results validates our approach on multi-resolution and distribution of scores.

For non-linear Elman-Net and LSTM, we use only one hidden layer initialized with our Temporal-SVM. The input PSDF are segmented under different sequence lengths: 20, 40 and 100. A 10% overlap is applied to promote smooth transition between neighboring video segments, minimizing the effects of initialization of RNN. The Elman-Net and LSTM are written in python with Theano library [3, 4], with smoothness and L_2 regularization, and ADADELTA [42] learning algorithm. The experiment results with Elman-Net and LSTM are shown in Table 3.

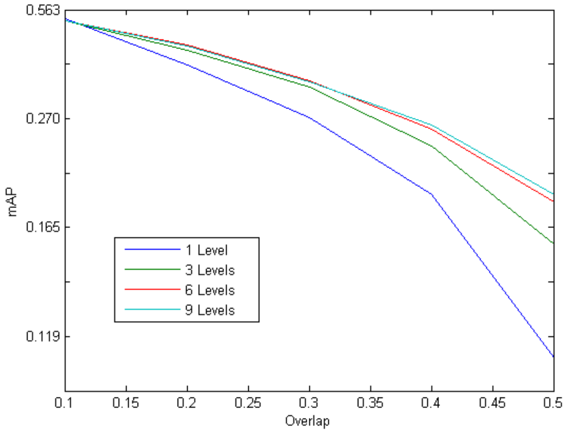

Figure 5. mAP Vs. # of Resolution Levels. The detection performance improves with a larger number of temporal scales. Log scale is used for visualizing mAP under larger overlaps.

mAP		0.1	0.2	0.3	0.4	0.5
Elman-Net	20	**62.1**	**51.0**	**38.0**	**26.4**	16.3
	40	60.8	49.2	36.8	26.1	16.5
	100	59.0	47.1	36.2	25.5	**16.9**
LSTM	20	58.8	47.5	35.9	24.9	16.0
	40	56.9	46.2	34.4	25.7	16.4
	100	55.1	43.9	33.9	24.8	16.3

Table 3. Detection performance with Elman-Net and LSTM network on PSDF with different sequence lengths. The context information is better utilized with the non-linear recurrent networks, resulting in enhanced detecion performance.

We also list the detection AP of each class in Figure 6. The easy classes include CleanAndJerk, CliffDiving and GolfSwing with relatively higher mAP, while CricketShot and FrisbeeCatch are difficult classes whose mAP are much lower.

5.2. MPII Cooking Activities Dataset

The MPII Cooking Activities Dataset [25] consists of high resolution videos of fine-grained activities. It has 65 classes of cooking activities with both classification and detection tasks. We use a similar experiment setting with THUMOS'15 except that the videos are not resized; this is because the fine-grained actions can only be well captured in high resolutions.

The midpoint hit criterion is used to evaluate the detection performance: if the midpoint of the detection lies in the ground truth, the detection is considered correct. Due to the computation cost of cross validation, we only evaluate the detection task on the first subject (also re-evaluation of the baseline results). The experiment results is shown in Table 4, where the utilization of RNN has greatly improved

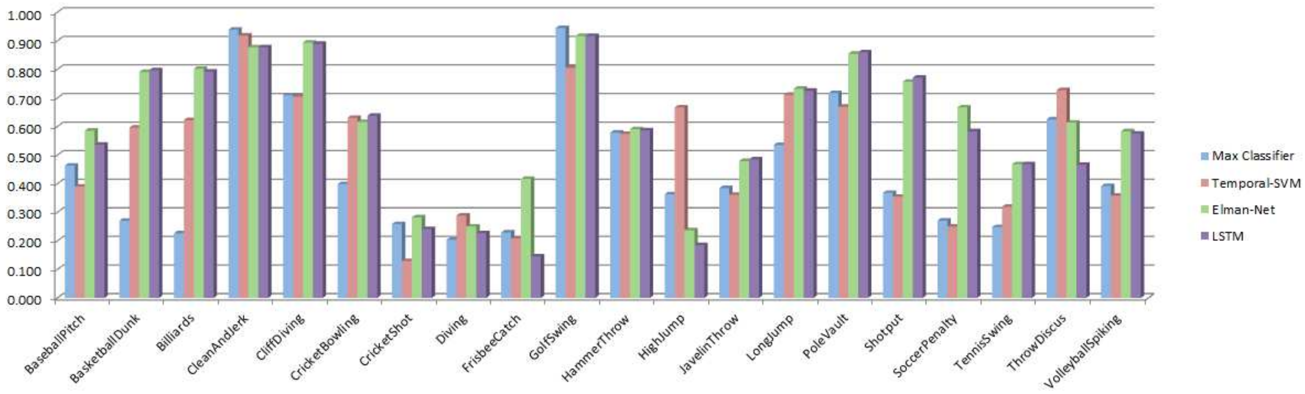

Figure 6. Detection AP over different action classes with overlap = 0.1.

the detection performance.

		Precision	Recall
Baseline [25]		17.7	40.3
Ours.	Temporal-SVM	27.3	42.5
	Elman-Net (20)	34.1	56.3
	LSTM (20)	**36.3**	**59.7**

Table 4. Detection results on MPII Cooking Dataset. Our experiments are based on the PSDF descriptor.

5.3. Computation Requirement

We use a server with Intel Core i7-5930K, 64 GB RAM, and an nVidia Geforce TITAN X graphics card for our experiments. The approximate computation requirement for THUMOS'15 Dataset and MPII Cooking Activities Dataset are shown in Table 5. Unlisted processes (dictionary, Temporal-SVM, etc.) have negligible computation cost. The computation of IDT takes most of the time, especially at high resolutions; it also consumes considerable disk space.

Process	Device	Storage	Time
Video	-	500G / 8G	-
Trajectory	cpu	10T / 240G	21d / 10d
Encoding	cpu	6T / 90G	20h / 4h
Class-SVM	cpu	-	1d / 1h
PSDF	gpu	40G / 350M	12h / 2h
Elman-Net	gpu	-	1d / 3h
LSTM	gpu	-	2d / 6h

Table 5. Computation requirement for THUMOS / MPII Datasets.

6. Future Studies

Our detection framework is based on trajectory features from actions. The studies in [39] use a novel feature pooling technique based on CNN as enhancement of trajectories, and achieve best performance in action classification in THUMOS'15. Incorporating scene features are more challenging in action localization tasks since they are more sensitive to the background, nonetheless it provides a novel study point we would like to further explore into.

We shall also study different network structures on the classification performance. The impact of different number of hidden layers, hidden units and feedback structures are to be further studied. It is also possible to learn feature descriptors at an earlier stage of processing.

The additive feature generation can be extended similarly in spatial domain. Combination with the heuristic information, like human / object detection and density of optical flows would provide prospects for more challenging spatial-temporal localizations.

7. Conclusion

We propose a novel PSDF descriptor for temporal action localization. Based on IDT, we apply Fisher Encoding on each anchor frame at multi-resolutions, followed by the Class-SVM to generate PSDF scores. The PSDF is a intuitive descriptor for action class, position and duration; and it can be calculated in an efficient linear recursive way. The descriptor is robust and representative by utilizing the distribution information.

We also apply the Elman-Net and LSTM networks to refine the localization tasks. These non-linear classifiers further exploit the consistency of context information and further boost the detection performance. We tested our framework on the THUMOS'15 and MPII Cooking Activities Dataset, and both give improved performance the previous attempts.

Acknowledgment: This work was done while Jun Yuan was on a research internship at Dr. Ni's group in SJTU. This work is partly supported by National Natural Science Foundation of China (Grant No.: 61502301).

References

[1] Lstm networks for sentiment analysis. http://deeplearning.net/tutorial/lstm.html.

[2] M. A. R. Ahad, J. Tan, H. Kim, and S. Ishikawa. Action dataseta survey. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 1650–1655. IEEE, 2011.

[3] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[4] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[5] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. June 2015.

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. 2014.

[7] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[9] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015.

[10] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015.

[11] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part V*, pages 3–20, 2014.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1725–1732, Washington, DC, USA, 2014. IEEE Computer Society.

[14] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[17] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[18] I. Laptev and P. Perez. Retrieving actions in movies. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[20] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[21] A. Patron, M. Marszalek, A. Zisserman, and I. Reid. High five: Recognising human interactions in tv shows. In *Proceedings of the British Machine Vision Conference*, pages 50.1–50.11. BMVA Press, 2010. doi:10.5244/C.24.50.

[22] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[23] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag.

[24] P. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In T. Jebara and E. P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 82–90. JMLR Workshop and Conference Proceedings, 2014.

[25] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1194–1201, June 2012.

[26] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[27] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360. ACM, 2007.

[28] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[30] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[31] A. Stoian, F. M., B.-P. J., and M. Crucianu. Action recognition and detection by combining motion and appearance fea-

tures. http://mexculture.cnam.fr/xwiki/bin/view/Datasets/Mex+action+dataset, 2014.

[32] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[33] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. *arXiv preprint arXiv:1504.06678*, 2015.

[34] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action Recognition by Dense Trajectories. In *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States, June 2011. IEEE.

[35] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

[36] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. http://crcv.ucf.edu/THUMOS14/papers/CUHK&SIAT.pdf, 2014.

[37] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Computer Vision–ECCV 2008*, pages 650–663. Springer, 2008.

[38] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *arXiv preprint arXiv:1504.01561*, 2015.

[39] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1798–1807, 2015.

[40] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 492–497. IEEE, 2009.

[41] J. Yuan, Z. Liu, and Y. Wu. Discriminative video pattern search for efficient action detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1728–1743, Sept 2011.

[42] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.