Transfer Latent SVM for Joint Recognition and Localization of Actions in Videos

Cuiwei Liu, Xinxiao Wu, and Yunde Jia, *Member, IEEE*

***Abstract*—In this paper, we develop a novel transfer latent support vector machine for joint recognition and localization of actions by using Web images and weakly annotated training videos. The model takes training videos which are only annotated with action labels as input for alleviating the laborious and time-consuming manual annotations of action locations. Since the ground-truth of action locations in videos are not available, the locations are modeled as latent variables in our method and are inferred during both training and testing phrases. For the purpose of improving the localization accuracy with some prior information of action locations, we collect a number of Web images which are annotated with both action labels and action locations to learn a discriminative model by enforcing the local similarities between videos and Web images. A structural transformation based on randomized clustering forest is used to map the Web images to videos for handling the heterogeneous features of Web images and videos. Experiments on two public action datasets demonstrate the effectiveness of the proposed model for both action localization and action recognition.**

***Index Terms*—Action localization, action recognition, transfer latent support vector machine (TLSVM) model.**

I. Introduction

ACTION recognition is one of the most active research topics in computer vision and plays an important role in wide applications such as intelligent video surveillance, content-based video retrieval, and human–computer interaction. In recent years, action recognition and localization have attracted extensive research interests, and some literature [1]–[4] engage in jointly predicting which action is performed (recognition) and where the action takes place (localization) in human action videos. However, most methods of action recognition and localization require both annotations of action classes and hand-labeling of regions of actions in each frame for training.

Manuscript received May 28, 2014; revised January 15, 2015 and April 29, 2015; accepted September 14, 2015. Date of publication October 15, 2015; date of current version October 13, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61203274 and Grant 61375044, in part by the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 20121101120029, in part by the Specialized Fund for Joint Building Program of Beijing Municipal Education Commission, and in part by the Excellent Young Scholars Research Fund of Beijing Institute of Technology. This paper was recommended by Associate Editor F. Karray. *(Corresponding author: Xinxiao Wu.)*

The authors are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: liucuiwei@bit.edu.cn; wuxinxiao@bit.edu.cn; jiayunde@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2482970

In order to alleviate the arduous and time-consuming manual annotations of action locations, we use a large number of training videos which are only annotated with action labels and a few of Web images which are annotated with both action labels and locations, to build a system for simultaneously recognizing and localizing actions. Specifically, spatiotemporal regions localizing the actions in videos are treated as latent variables, and the best candidate regions are automatically selected in both training and test videos. In the training stage, the Web images are introduced for compelling the spatiotemporal regions of interest from training videos to resemble the annotated regions of interest from Web images, which benefits improving the localization accuracy.

Some recent literature [5]–[7] also consider to localize and recognize actions using training videos only annotated with action labels. These methods create candidate spatiotemporal regions without supervision and take one or more spatiotemporal regions discriminative for action recognition as the results of action localization. These methods assume that the most discriminative parts of videos are actually the spatiotemporal regions of the actions. However, for many actions such as diving and golf swing, instances usually share similar scenarios. Consequently, regions of background are more discriminative than regions of motions for action recognition, which would lead to incorrect localizations.

We propose a novel transfer latent support vector machine (TLSVM) for jointly recognizing and localizing actions in videos by using Web images and weakly annotated training videos. The model takes the spatiotemporal regions of actions as latent variables and selects the best one from a set of region candidates in both training and test videos. During the training stage, the local similarities between spatiotemporal regions of interest from training videos and the annotated regions of interest from Web images are enforced to boost both action recognition and localization. At test time, the proposed model is able to automatically predict both action label and location in an input video. In this paper, bag-of-words representations based on randomized clustering forest are adopted to characterize videos and Web images. Since videos and Web images are represented by heterogeneous features generated from different code books, we introduce a structural transformation based on randomized clustering forest to transform the image feature space to the video feature space. Furthermore, we utilize an unsupervised method to yield a set of spatiotemporal region candidates for each video, in which salient regions from all video frames are grouped into each region candidate by a two-stage algorithm based on

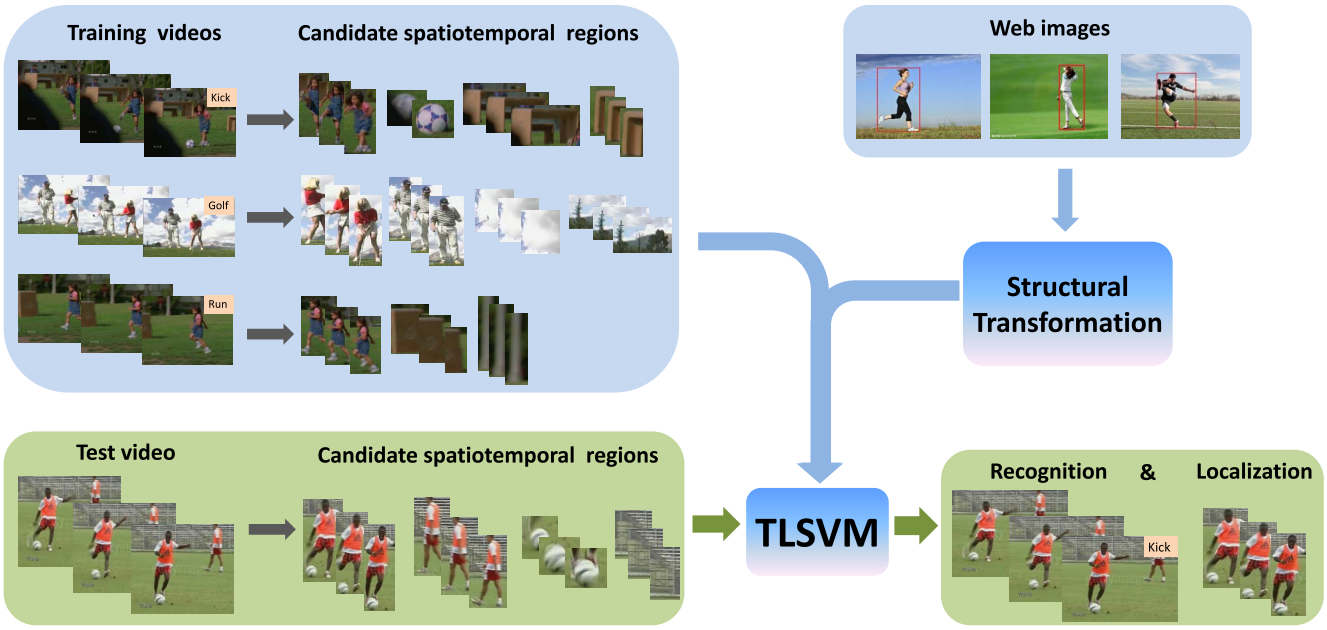


Fig. 1. Overview of the proposed method. TLSVM is trained on a set of weakly annotated training videos and a few Web images annotated with both action labels and action locations. A structural transformation is learned to map the feature space of images to the feature space of videos. For each video, a set of candidate spatiotemporal regions is created using a novel unsupervised method. During test, the model is able to automatically predict both action label and action location for an input video.

affinity propagation clustering. An overview of our approach is illustrated in Fig. 1.

The rest of this paper is organized as follows. Section II summarizes the related work. In Section III, we describe the representation of videos and Web images, including the bag-of-words framework based on randomized clustering forest and the structural transformation from images to videos. Section IV elaborates on the generation of candidate spatiotemporal regions. We then present the TLSVM model for action recognition and localization in Section V. Experimental results and the conclusion are reported in Sections VI and VII, respectively.

II. Related Work

A. Action Recognition and Localization

A lot of existing action recognition methods [8]–[14] concentrate on determining which action occurs in an input video, regardless of where the action really takes place. However, in many cases it is desirable to know where the action occurs as well as to recognize the action being performed.

Recently, the problem of simultaneously recognizing and localizing actions from videos has been widely studied. Oikonomopoulos *et al.* [1] proposed an implicit representation of the spatiotemporal shape of an activity for localizing and recognizing human actions in unsegmented image sequences. In their work, the upper and lower bounds of the subjects are manually annotated at each frame. Yao *et al.* [2] presented an approach to classify and localize actions using a Hough transform voting framework. They annotated each frame of training examples with a bounding box, in order to obtain normalized action tracks to build a hough forest. Lan *et al.* [3] formulated a discriminative model coupling action recognition with person localization. Although this method utilizes a latent region of interest to indicate the action location, it still requires the supervision of latent region in each frame of training videos. Raptis *et al.* [4] focused on discovering discriminative action parts from clusters of local trajectories that are densely sampled from the videos for action recognition and localization. In their work, strongly supervised bounding boxes of all training frames are extracted to restrict the selection of action parts. Tian *et al.* [15] proposed a spatiotemporal deformable part model (SDPM) by extending the deformable part model [16] from 2-D images to 3-D spatiotemporal volumes for detecting actions in videos. In SDPM, ground-truth of action locations in training videos is required to learn global root filter of each action. Cao *et al.* [17] developed an adaptive cross-dataset action detection approach, which explores the spatiotemporal coherence of actions and incorporates the prior information for cross-dataset analysis. In their work, locations of actions in source dataset are provided to train an original action classifier. Yao *et al.* [18] proposed animated pose templates for detecting short-term, long-term, and contextual actions in real-world videos. They employed training videos with annotated bounding boxes to learn shape templates. Amer and Todorovic [19] presented a generative chain model to recognize and localize activities of a group of people in videos. All of the aforementioned approaches require the manual annotation of action location for each frame as well as the action label for the entire video.

Some recent literature have targeted the problem of action localization and recognition using training videos only annotated with action labels. Shapovalova *et al.* [5] proposed a similarity constrained latent SVM model for weakly supervised

action recognition and localization. This model aims to advance the recognition performance by enforcing the consistency of local regions among training data, and uses the regions that are most discriminative for recognition as localization results. Ma *et al*. [6] presented to generate hierarchical space-time segments in an unsupervised manner, and these segments are utilized as the action representation for classification. In their work, localization of the action is achieved by outputting space-time segments that have positive contributions to the classification. Boyraz *et al*. [7] developed an action recognition system that automatically locates discriminative regions in a video and then utilizes information from these regions for action classification. The focus of their method is to find the most discriminative regions for action classification, and they assumed that the regions containing the actual action tend to be more discriminative than the others. However, in many cases, a region from the background may be chosen as the action localization result due to the similar scenarios shared among training videos with the same action label. Our approach conquers this problem by introducing Web images which are annotated with both action labels and action locations. Local similarities between spatiotemporal regions of interest from training videos and annotated regions of interest from Web images are enforced to boost both action recognition and localization.

B. Transferring Knowledge From Images to Videos

Duan *et al*. [20] developed a multiple source domain adaptation method for event recognition in consumer videos by leveraging a large number of Web images from different sources. Chen *et al*. [21] proposed an event recognition model for consumer videos, using a large number of loosely labeled Web videos and Web images. Chen *et al*. [22] designed an automatic semantic concept discovery scheme for recognizing complex events in videos, by exploiting Internet images and their associated tags. All of these methods focus on event recognition without considering the localization task, while the proposed approach can simultaneously recognize and localize the action in a video.

Ikizler-Cinbis *et al.* [23] and Ikizler-Cinbis and Sclaroff [24] employed action pose classifiers trained with a large image dataset to detect actions in each frame of an input video. A key difference between our approach and their methods is that their methods focus on transferring knowledge from images to images, while our model is able to transfer knowledge from Web images to videos for recognizing and localizing actions in videos.

III. Representation of Videos and Images

This paper aims to address the problem of joint action recognition and localization in videos. In order to alleviate the laborious and time-consuming manual annotations of action locations, we present a novel TLSVM model which is learned using Web images and weakly annotated videos. The proposed model treats spatiotemporal regions of actions in videos as latent variables, and automatically selects the best one from a set of region candidates by enforcing the local similarities between training videos and Web images.

In this section, we first describe how to represent videos and Web images in a bag-of-words framework based on randomized clustering forest [25]. Then a structural transformation for mapping the image feature space to the video feature space is presented since videos and images are represented by heterogeneous features.

A. Bag-of-Words Representation Based on Randomized Clustering Forest

Bag-of-words model [26] is a popular and powerful method for classification and recognition, which quantizes the low-level local descriptors as a histogram of visual words to get a discriminative mid-level representation. In [25], randomized clustering forest is presented for building the visual code-book of bag-of-words model, in which each leaf node is regarded as a separate visual word. In this paper, we use the randomized clustering forest [25] to quantize low-level descriptors effectively.

Web images are characterized by a set of densely sampled low-level histograms of oriented gradient (HOG) descriptors [27] $\{z_l^{\text{HOG}}\}_{l=1:N_I}$, and videos are described by dense trajectories [28] $\{z_k^{\text{traj}}\}_{k=1:N_V}$. For trajectory k, a descriptor z_k^{traj} is extracted within a space-time volume around the trajectory, and an HOG descriptor z_k^{HOG} is extracted to characterize the spatial patch. The trajectory descriptors $\{z_k^{\text{traj}}\}_{k=1:N_V}$ are utilized to construct the randomized clustering forest for videos, while two sets of HOG descriptors $\{z_k^{\text{HOG}}\}_{k=1:N_V}$ and $\{z_l^{\text{HOG}}\}_{l=1:N_I}$ are integrated to build the randomized clustering forest for images. Moreover, the correspondence between z_k^{traj} and z_k^{HOG} are exploited to learn a transformation from images to videos, which will be described in detail in Section II-B.

Randomized clustering forest is an ensemble of decision trees, and the tree hierarchies provide a means of clustering low-level local descriptors. Nodes of each tree constitute the hierarchical clusters, namely, the visual words in bag-of-words model. Histograms of visual words in videos are generated from clustering forests built upon trajectory descriptors, while histograms of visual words in images are created from forests built upon HOG descriptors.

1) Construction of Trees: Each tree in a clustering forest is independently grown from a random subset D' of the labeled training low-level descriptors D in a top-down manner. We assume that low-level descriptors share the same label with the video or image they are sampled from. All the training data in D' are dropped down from the root of a tree. In order to split a node n, we generate a set of N_H hypotheses $\{(c_k^n, t_k^n)_{k=1:N_H}\}$ randomly, where c_k^n denotes one feature candidate and t_k^n is the corresponding threshold for splitting. Each hypothesis divides the training data arriving at the node n into two subsets, and the one maximizing the expected information gain is chosen for node split. Growth of a tree is controlled by a maximum tree depth and a minimum amount of samples, so a node stops splitting in the following three cases: 1) the limited tree depth is reached; 2) there are not enough data for splitting; and 3) all

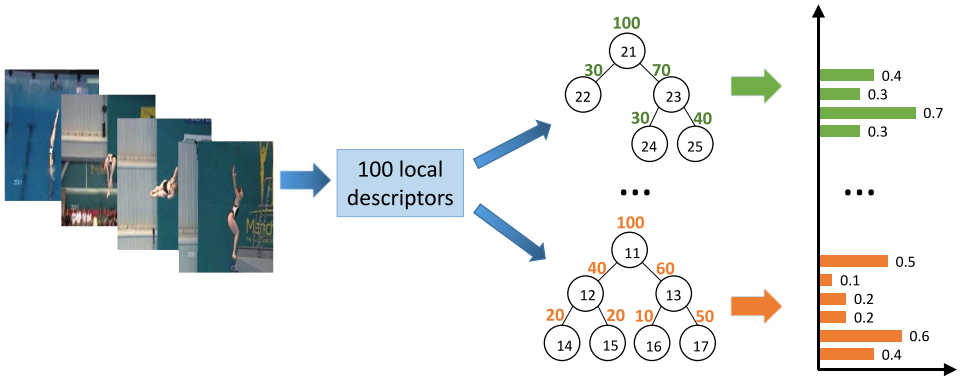


Fig. 2. Generating the mid-level representation for a video.

the data belong to the same class. If one of the above three conditions is satisfied, the node will be treated as a leaf.

2) Data Coding: We take all the nodes (except the root) of each tree, including split nodes and leaf nodes as hierarchical visual words in our framework. Randomized forests on videos and images are built separately by their corresponding training low-level descriptors, we quantize the visual words for videos and images in the same way. Taking a video for example, all the extracted local trajectory descriptors are dropped down from the root of each tree, and the occurrences of nodes across all trees are concatenated to create a normalized histogram H, as shown in Fig. 2. Suppose $\mathbf{H}(n)$ to be the occurrence of split node n, then $\mathbf{H}(n)$ can be calculated as

$$\mathbf{H}(n) = \mathbf{H}(n_L) + \mathbf{H}(n_R) \tag{1}$$

where n_L and n_R denote the left and right children nodes of node n, respectively. The hierarchical histogram encodes the structure of each tree, and the relationship among father node and children nodes [defined in (1)] is employed to learn a linear transformation in the next section.

B. Structural Transformation

In order to cope with the heterogeneous features of images and videos, a class specific structural transformation is introduced to map the image feature space to the video feature space.

Assume that $\mathrm{RF}^V = \{T_r^V\}_{r=1:N_T}$ and $\mathrm{RF}^I = \{T_r^I\}_{r=1:N_T}$ are randomized clustering forests on videos and images, respectively, where T_r denotes the rth tree in a forest and N_T is the number of trees. Training trajectory descriptors of videos $\{z_k^{\mathrm{traj}}\}_{k=1:N_V}$ are passed through T_r^V from the root, and the corresponding HOG descriptors $\{z_k^{\mathrm{HOG}}\}_{k=1:N_V}$ are dropped down to T_r^I simultaneously.

We first learn a set of class specific mapping matrices $\{\mathbf{L}_r^y\}_{y\in Y} \in R^{\mathrm{Nl}_r^V \times \mathrm{Nl}_r^I}$ among the leaf nodes of T_r^I and T_r^V by using the correspondence between low-level descriptors, where Nl_r^V is the number of leaf nodes in tree T_r^V, and Nl_r^I is the number of leaf nodes in tree T_r^I. Each element $\mathbf{L}_r^y(p,q)$ in matrix $\mathbf{L}_r^y$ is obtained by calculating the amount of samples k of action y, that z_k^{traj} reaches leaf node p of tree T_r^V and z_k^{HOG} goes to leaf node q of tree T_r^I. Normalization is performed on each column of $\mathbf{L}_r^y$ afterwards.

Suppose $\mathbf{H}_r^I$ to be the histogram of an image with action label y, generated by tree T_r^I, and $\mathbf{Hl}_r^I \in R^{\mathrm{Nl}_r^I \times 1}$ to be a sub-histogram of $\mathbf{H}_r^I$ corresponding to leaf nodes, we can get a transformed sub-histogram $\mathbf{Hl}_r^V \in R^{\mathrm{Nl}_r^V \times 1}$ by defining each element in $\mathbf{Hl}_r^V$ as

$$\mathbf{Hl}_r^V(p) = \sum_{q=1:Nl}^{I_r} \mathbf{L}_r^y(p,q) \cdot \mathbf{Hl}_r^I(q). \tag{2}$$

With the transformed sub-histogram $\mathbf{Hl}_r^V$ of leaf nodes, we can create the transformed histogram $\mathbf{H}_r^V$ of all nodes according to (1).

Since both of the transformations defined by (1) and (2) are linear, the whole transformation from $\mathbf{H}_r^I$ to $\mathbf{H}_r^V$ is also a linear transformation. Transformed histograms of all trees $\{\mathbf{H}_r^V\}_{r=1:N_T}$ are concatenated to form the transformed mid-level representation of the Web image. In the following, we use a matrix $\mathbf{A}$ to represent the linear transformation from the feature space of images to that of videos, for convenience.

IV. Spatiotemporal Regions of Interest

We address the problem of jointly recognizing and localizing the action in an input video, that means the proposed model could provide a spatiotemporal region of interest capturing where and when the action occurs in addition to predicting the action label. Given an input video, a set of potential regions of interest is generated in advance, and our model automatically selects a best candidate during learning and inferring. Therefore, how to produce a reasonable and reduced set of candidate spatiotemporal regions of interest for each video is of great importance.

Generally, a spatiotemporal region of interest within a video could be formed in two manners. One intuitive strategy is to extract global 3-D bounding boxes covering the whole action. However, this constrained structure is only applicable for actions with stable locations in a video (i.e., boxing and handshake), and does not work well on drastic actions such as running and walking. Another alternative resorts to independently extracting 2-D bounding boxes from each frame to constitute reasonable regions of interest for an input video, and we adopt this scheme in this paper. In the rest of this section, we describe the procedures of

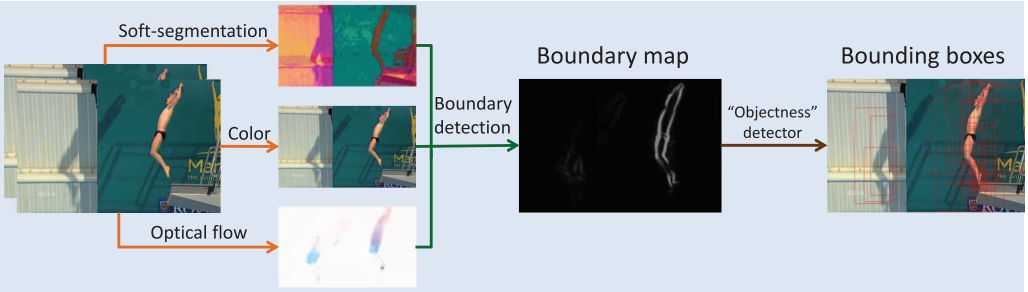


Fig. 3. Illustration of extracting bounding boxes from a frame.

extracting bounding boxes from each frame, refining and pruning the bounding boxes, and generating potential regions of interest.

A. Extracting Bounding Boxes

Given an input video, an "objectness" detector [29] is utilized to extract bounding boxes that are likely to contain an object of interest on each frame of the video. The objectness detector is performed on static color images, without considering motion information. However, motion information such as optical flow can be complementary to the static appearance information for action recognition of image sequences. Appearance information characterizes the static pattern of an image, while motion information captures the focus of action and allows to discard some irrelevant parts from the background. For this reason, we introduce motion information into the extraction of bounding boxes, instead of performing objectness detector directly on the original color images.

In practice, we utilize the approach in [30] to detect boundaries for each frame merging both appearance information of a single image and the motion information between adjacent frames. Six appearance channels (i.e., color and soft-segmentation [30]) and two optical flow [31] channels are combined to compute the boundary map for each frame. Then the objectness detector operates on the boundary maps and returns the potential bounding boxes. In consideration of the variety of human poses in action videos, the objectness detector adopts a low threshold to output sufficient bounding boxes. It is notable that the motion information inhibits the salience of static objects in boundary maps, facilitating the objectness detector to filter irrelevant objects from background. Fig. 3 illustrates the process of extracting bounding boxes from a frame with appearance and motion information.

B. Refining and Pruning Bounding Boxes

Due to the low threshold employed by the objectness detector, we got abundant bounding boxes from each frame. However, many of the output bounding boxes are larger than the actual objects, and there is also much overlap between some bounding boxes within a single frame. Therefore, a refining and pruning process is needed to produce a concise set of precise bounding boxes.

To this end, we first refine each candidate bounding box with an adaptive threshold ϱ by removing the outer part with small boundary intensity. Suppose the boundary map within a bounding box is $\mathbf{BM}$, then we remove the leftmost N_{left} columns, the rightmost N_{right} columns, the top N_{top} rows, and the bottom N_{top} rows of $\mathbf{BM}$, that the boundary of each point in these columns and rows is lower than ϱ. The adaptive threshold ϱ is given by

$$\varrho = \frac{\sum_{i,j} \mathbf{BM}(i,j)}{\sum_{i,j} \mathbb{1}(\mathbf{BM}(i,j) > 0)}$$

where $\mathbb{1}(\mathbf{BM}(i,j) > 0)$ is an indicator function, that is $\mathbb{1}(\mathbf{BM}(i,j) > 0)$ is 1 if $\mathbf{BM}(i,j) > 0$ and 0 otherwise.

Then we prune redundant bounding boxes by merging the boxes with large common area. For each pair of bounding boxes (B_i, B_j) extracted from the same frame, we compute their intersection-over-union (IOU) score as $(\text{Area}(B_i \cap B_j))/(\text{Area}(B_i \cup B_j))$. If the IOU score is larger than a threshold (in our experiments, it is set to be 0.7), we regard the two bounding boxes as largely overlapped and only reserve the box with larger objectness score. By reducing redundant bounding boxes, we are able to obtain a condensed set of candidate bounding boxes.

C. Generating Candidate Regions of Interest

The goal of this section is to construct a set of candidate regions of interest for an input video with the concise set of bounding boxes extracted from each frame. Intuitively, a spatiotemporal region of interest can be regarded as a group of bounding boxes from different frames, thus we develop a two-stage algorithm based on affinity propagation [32] to group the bounding boxes into different spatiotemporal regions of interest.

In the first stage, we utilize affinity propagation cluster algorithm to group the bounding boxes into hundreds of clusters based on their appearance similarities and spatiotemporal distances. Affinity propagation is an exemplar based cluster algorithm, taking a similarity matrix between samples as input. Intuitively, bounding boxes that are both similar in appearance and adjacent in space and time fall in the same cluster. In order to group bounding boxes, we need to define a similarity measure between bounding boxes. Given two bounding boxes $B_i = (\mathbf{h}_i, \mathbf{a}_i, \mathbf{c}_i, \mathbf{t}_i)$ and $B_j = (\mathbf{h}_j, \mathbf{a}_j, \mathbf{c}_j, \mathbf{t}_j)$, where $\mathbf{h}_i$ is the color histogram, $\mathbf{a}_i$ denotes the area, $\mathbf{c}_i$ denotes the spatial coordinates for the center point, and $\mathbf{t}_i$ represents the

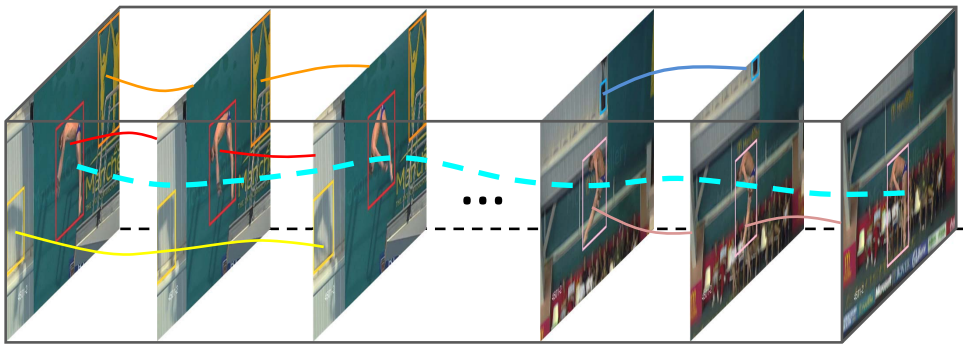


Fig. 4. Illustration of generating spatiotemporal regions of interest for a video. The actual lines represent first-stage clusters, and the aqua dotted line indicates a second-stage cluster. We can see that first-stage clusters are composed of similar bounding boxes from adjacent frames, and the red and pink first-stage clusters are grouped into a second-stage cluster to create a spatiotemporal region of interest.

temporal coordinate. We define the similarity measure of B_i and B_j as

$$\begin{aligned} S_B(B_i, B_j) &= -w_h \cdot \mathcal{D}_h(\mathbf{h}_i, \mathbf{h}_j) - w_a \cdot \mathcal{D}_a(\mathbf{a}_i, \mathbf{a}_j) \\ &\quad - w_s \cdot \mathcal{D}_s(\mathbf{c}_i, \mathbf{c}_j) - \mathcal{D}_t(\mathbf{t}_i, \mathbf{t}_j) \\ w_a &= \varsigma_a \cdot \exp\left(-|\mathbf{t}_i - \mathbf{t}_j|\right) \\ w_s &= \varsigma_s \cdot \exp\left(-|\mathbf{t}_i - \mathbf{t}_j|\right) \\ w_h &= 1 - w_a - w_s \end{aligned} \tag{3}$$

where $\mathcal{D}_h$, $\mathcal{D}_a$, $\mathcal{D}_s$, and $\mathcal{D}_t$ denote the χ^2 distance between two color histograms, the difference between the area, the spatial Euclidean distance between two center points and the temporal distance between two bounding boxes, respectively. w_h, w_a, w_s are the corresponding weights. In (3), as the difference between $\mathbf{t}_i$ and $\mathbf{t}_j$ increases, both weight for area (i.e., w_a) and weight for spatial distance (i.e., w_s) decrease, while the weight for color histogram w_h increases. That means, for temporally distant bounding boxes, the appearance similarity (i.e., similarity for color histograms) accounts for a larger proportion of S_B. In contrast, for temporally adjacent bounding boxes, the area similarity and spatial similarity are more important.

Due to the temporal distance $\mathcal{D}_t$, bounding boxes extracted from temporally distant frames will fall into different clusters, and each cluster is composed of similar bounding boxes from adjacent frames. Small clusters are discarded, because bounding boxes within them are likely to be extracted from the background. We iterate the above process to eliminate irrelevant background bounding boxes.

In the second stage, affinity propagation cluster algorithm is performed on the first-stage clusters, according to the similarities between bounding boxes in different first-stage clusters. This results in tens of second-stage clusters, and bounding boxes appear in the same second-stage cluster form a spatiotemporal region of interest. The similarity between two first-stage clusters C_k^1 and C_l^1 is defined as

$$\begin{aligned} S_C\left(C_k^1, C_l^1\right) = \max_{i,j:B_i \in C_k^1, B_j \in C_l^1} & -w_h \cdot \mathcal{D}_h(\mathbf{h}_i, \mathbf{h}_j) \\ & - w_a \cdot \mathcal{D}_a(\mathbf{a}_i, \mathbf{a}_j) - w_s \cdot \mathcal{D}_s(\mathbf{c}_i, \mathbf{c}_j). \end{aligned} \tag{4}$$

Different from (3), the similarity measure in (4) does not take the temporal distance $\mathcal{D}_t$ of bounding boxes into consideration. Accordingly, similar bounding boxes from adjacent frames are grouped into clusters in the first stage, and then distant first-stage clusters with similar appearances are allowed to be clustered together in the second stage. Finally, we discard small second-stage clusters which are likely to be background bounding boxes, and each second-stage cluster with sufficient bounding boxes forms a region of interest for the video. Fig. 4 depicts a spatiotemporal region of interest.

V. Transfer Latent SVM Model

The TLSVM model is able to predict both which action happens and where this action locates in an action video. A few Web images annotated with both action labels and action locations are employed to learn a discriminative model. The model does not require the manual annotation of action locations in training videos, and could automatically choose a latent region of interest within each video, which significantly reduces the labeling complexity.

A. Model Formulation

Let $\mathcal{D}^V = \{(x_i, y_i)_{i=1:N}\}$ be the training videos, where $y_i \in Y$ is the action label of video x_i, and the unobserved action locations $\{h_i\}_{i=1:N}$ of videos are treated as latent variables in our model. The latent variable h_i specifies a local spatiotemporal region in video x_i. Our method aims to learn a discriminative compatibility function $F(x, y)$ which measures how compatible the action label y is suited to an input video x

$$\begin{aligned} F(x, y) &= \max_h f_\omega(x, y, h) \\ f_\omega(x, y, h) &= \omega^T \Phi(x, y, h) \end{aligned}$$

where ω is the learned parameter of the model, and $\Phi(x, y, h)$ is a joint feature vector which describes the relationship among the action video x, the action label y, and the latent action location h.

The model parameter includes two parts $\omega = \{\alpha; \beta\}$. The relationship among an action video x, an action label y and the latent region h is formulated as

$$\begin{aligned} \omega^T \Phi(x, y, h) &= \alpha^T \varphi_1(x, y) + \beta^T \varphi_2(x, h, y) \\ \alpha^T \varphi_1(x, y) &= \sum_{t=1}^{N_y} \alpha_t^T \cdot \phi(x) \cdot \mathbb{1}(y = t) \\ \beta^T \varphi_2(x, h, y) &= \sum_{t=1}^{N_y} \beta_t^T \cdot \psi(x, h) \cdot \mathbb{1}(y = t) \end{aligned} \tag{5}$$

where $\mathbb{1}(y=t)$ is an indicator function, with $\mathbb{1}(y=t)=1$ if $y=t$ and 0 otherwise. In (5), the potential function $\alpha^T\varphi_1(x,y)$ captures the global relationship between an action video x and the action label y, where $\phi(x)$ denotes a mid-level representation obtained by the random clustering forest using low-level trajectory descriptors extracted from the whole video. The potential function $\beta^T\varphi_2(x,h,y)$ measures the compatibility between a local region h and the action label y, where $\psi(x,h)$ is also a mid-level feature vector, but only using low-level trajectory descriptors extracted from a local region of x specified by the latent variable h.

B. Learning

Given a set of weakly labeled training videos $\mathcal{D}^V=\{(x_i,y_i)_{i=1:N}\}$ and a few Web images $\mathcal{D}^I=\{(x_j^I,y_j^I,h_j^I)_{j=1:M}\}$, where $y_j^I\in Y$ is the action label of image x_j^I and h_j^I indicates the spatial location of the person, our goal is to learn the model parameter ω. Since the unobserved action locations of training videos $\{h_i\}_{i=1:N}$ are treated as latent variables, the model is formulated in a latent structural SVM framework for learning

$$\min_{\omega,\xi_i,\xi_j^I,\xi_i^s}\ \frac{1}{2}\|\omega\|^2+C_1\sum_{i=1}^{N}\xi_i+C_2\sum_{j=1}^{M}\xi_j^I+C_3\sum_{i=1}^{N}\xi_i^S \quad (6)$$

$$\text{s.t.}\quad f_\omega(x_i,y_i,h_i)-f_\omega\big(x_i,y',h'\big)\geq\Delta\big(y_i,y'\big)-\xi_i;\ \forall y',\forall h',\forall i \quad (7)$$

$$g_\omega\Big(x_j^I,y_j,h_j\Big)-g_\omega\Big(x_j^I,y',h_j\Big)\geq\Delta\big(y_j,y'\big)-\xi_j^I;\ \forall y',\forall j \quad (8)$$

$$\min_{j:y_i=y_j}\frac{1}{Z_{x_i}}\cdot\Theta\Big((x_i,h_i),\Big(x_j^I,h_j\Big)\Big)\leq\xi_i^S \quad (9)$$

where ξ_i and ξ_i^S are slack variables for training video x_i, and ξ_j^I is the slack variable for Web image x_j^I. The normalization factor Z_{x_i} for video x_i is defined by

$$Z_{x_i}=\max_h\min_{j:y_i=y_j}\Theta\Big((x_i,h),\Big(x_j^I,h_j\Big)\Big). \quad (10)$$

Equation (7) represents the usual latent SVM max margin constraints which optimize ω by classifying training videos correctly. The loss function $\Delta(y,y')$ measures the cost of predicting the truth label y as action label y'. We define $\Delta(y,y')$ as a simple Hamming loss: $\Delta(y,y')$ is 1 if $y\neq y'$ and 0 otherwise.

Equation (8) denotes the max margin constraints for the transferred Web images. The constraints defined in (7) and (8) compel the model to classify both Web images and training videos. Different from the training videos, the Web images are annotated with the regions of actions, therefore (8) does not include any latent variables. $g_\omega(x^I,y,h)$ is the score function for Web images, defined by

$$g_\omega\big(x^I,y,h\big)=\sum_{t=1}^{N_y}\alpha_t^T\cdot\mathbf{A}\cdot\phi\big(x^I\big)+\sum_{t=1}^{N_y}\beta_t^T\cdot\mathbf{A}\cdot\psi\big(x^I,h\big)$$

where $\mathbf{A}$ is a learned mapping matrix transforming the image feature space to the video feature space, as Web images and videos are represented by heterogeneous features with different dimensions.

Equation (9) enforces the local similarities between training videos and Web images, which means that the latent regions of training videos should resemble the regions of actions annotated in Web images. According to this constraint, TLSVM model is inclined to choose latent regions with more similarity or less distance to the annotated local regions of images, which benefits both classification and localization. Here we define the loss function $\Theta((x_i,h_i),(x_j^I,h_j^I))$ as a pair-wise distance to estimate the similarity between a local region of image and a latent region of video, which can be directly calculated using mapping matrix $\mathbf{A}$ as

$$\Theta\Big((x_i,h_i),\Big(x_j^I,h_j^I\Big)\Big)=d\Big(\psi(x_i,h_i),\mathbf{A}\cdot\psi\Big(x_j^I,h_j^I\Big)\Big).$$

A variety of distance functions can be employed to measure the similarity between a video and an image, and we adopt the χ^2 distance which is suitable for histogram similarity estimation.

The optimization problem in (6) is non-convex since the latent variables $\{h_i\}_{i=1:N}$ are not observed during learning. Therefore we employ the non-convex bundle optimization algorithm [33]. In a nutshell, this algorithm iteratively builds a gradually accurate piecewise quadratic approximation to the objective function. At each iteration, calculation of the subgradient is required to add a new linear cutting plane to the piecewise quadratic approximation.

The objective function in (6) can be rewritten in an unconstrained form

$$O(\omega)=\min_\omega\frac{1}{2}\|\omega\|^2+\sum_{i=1}^{N}(L_i-R_i)+\sum_{j=1}^{M}P_j^I \quad (11)$$

where L_i, R_i, and P_j^I are defined by

$$L_i=C_1\max_{y',h'}\big[f_\omega\big(x_i,y',h'\big)+\Delta\big(y',y_i\big)\big]$$

$$R_i=\max_{h_i}\left[C_1f_\omega(x_i,y_i,h_i)-\frac{C_3}{Z_{x_i}}\min_j\Theta\Big((x_i,h_i),\Big(x_j^I,h_j\Big)\Big)\right]$$

$$P_j^I=C_2\left\{\max_{y'}\Big[g_\omega\Big(x_j^I,y',h_j^I\Big)+\Delta\big(y',y_j\big)\Big]-g_\omega\Big(x_j^I,y_j,h_j^I\Big)\right\}.$$

Assume that (y_i^*,h_i^*), h_i, and y_j^* are solutions to L_i, R_i, and P_j^I, respectively, the subgradient of $O(\omega)$ in (11) can be calculated by

$$\partial_\omega(O(\omega))=C_1\sum_{i=1}^{N}\big(\Phi\big(x_i,y_i^*,h_i^*\big)-\Phi(x_i,y_i,h_i)\big)+C_2\sum_{j=1}^{M}\Big(\Phi\Big(x_j^I,y_j^*,h_j^I\Big)-\Phi\Big(x_j^I,y_j,h_j^I\Big)\Big).$$

We enumerate y', h', and h_i to find the optimal (y_i^*,h_i^*), h_i, and y_j^*.

C. Inference

With the learned parameter ω, the inference problem is to simultaneously find the best action label y^* and the best latent

region h^* given an input video x. The inference is equal to the following optimization problem:

$$\left(y^*, h^*\right) = \arg\max_{y,h} \omega^T \Phi(x, y, h). \tag{12}$$

We can solve (12) by enumerating all the possible action labels y and latent regions h for a test video x, as the set of possible values for y and h is limited.

VI. Experiments

A. Human Action Datasets

To evaluate the effectiveness of our method, we conduct experiments on the University of Central Florida (UCF) sports dataset [34] and the Olympic sports dataset [35].

1) UCF Sports Dataset: This dataset contains 150 sports videos of ten different human actions: 1) diving; 2) golf swing; 3) kicking; 4) lifting; 5) horse-riding; 6) running; 7) skating; 8) swing bench; 9) swing side; and 10) walking. Videos in this dataset are extracted from sports broadcasts, and bounding boxes of the person performing the action are provided for each frame. For fair comparison, we adopt the test strategy proposed in [3], which divides the dataset by choosing one third of the videos as the test data, leaving the rest for training.

2) Olympic Sports Dataset: This dataset is composed of 783 sports videos of 16 action classes: 1) *Basketball*; 2) *Bowling*; 3) *Clean-Jerk*; 4) *Discus-throw*; 5) *Diving-platform*; 6) *Diving-springboard*; 7) *Hammer-throw*; 8) *High-jump*; 9) *Javelin-throw*; 10) *Long-jump*; 11) *Pole-vault*; 12) *Shot-put*; 13) *Snatch*; 14) *Tennis-serve*; 15) *Triple-jump*; and 16) *Vault*. Complex sports actions, drastic camera motions, poor light and large variations of human appearance augment the difficulty of both action recognition and localization. The whole dataset is split into 649 videos for training and 134 videos for testing. We annotate the Olympic sports dataset with bounding boxes in order to quantify our localization performance.

3) Web Images: We use Google Image Search Engine to download images from the Web taking the action class labels as query keywords, and annotate a bounding box around the person of interest for each image. During training, five labeled Web images of each action are employed to learn our model. Examples of the Web images for the UCF sports dataset and the Olympic sports dataset are shown in Figs. 5 and 6, respectively.

B. Experimental Setting

In our implementation, HOG and MBH descriptors of dense trajectory [28] are extracted from videos, and HOG descriptors are densely sampled from the Web images. We randomly select 100 000 training descriptors to build the clustering forests for videos and images. We build clustering forests with different parameters (i.e., number of trees and tree depth), and adopt out-of-bag estimate [36] to assess their performance. Fig. 7 depicts the performance of clustering forests on the UCF sports dataset. As shown in Fig. 7, the curves increase first and then decrease with the tree depth due to the overfitting problem. It is also observed that the performance of clustering forest improves with the increasing number of trees. In practice, we should consider both performance and computation cost. Therefore we set the number of trees to five in our experiments, and choose the tree depth with the highest performance of out-of-bag estimate. Particularly, the depth of each tree in the clustering forest on Web images is limited to twelve, and the tree depth of the clustering forest on videos is limited to 16 and 11 for the UCF sports dataset and the Olympic sports dataset, respectively. For the generation of spatiotemporal regions, we employ a set of parameters $(\varsigma_a, \varsigma_s) \in \{(0.3, 0.3), (0.4, 0.4), (0.2, 0.6), (0.1, 0.7)\}$ to adapt to the variety of action videos.

We compare the proposed approach with three baseline methods.

1) Global Linear SVM Model Without Images: It only considers the first potential function $\alpha^T \varphi_1(x, y)$ in (5), which captures the global relationship between a video x and the action label y. All the trajectory descriptors extracted from a video are passed through each tree in the randomized clustering forest to create a mid-level bag-of-words representation of the whole video. A linear SVM classifier is trained on the global representations of training videos. Note that this method can only assign an action label to a test video, without predicting the location of person.

2) Latent SVM Model Without Images: It is similar to our method, except that no Web images are employed. Regions of interest are also treated as latent variables, but the local similarities between training videos and Web images are not enforced in this model. Particularly, only the parameter ω under the constraint in (7) is optimized, and the constraints in (8) and (9) are neglected.

3) TLSVM Model Using Frames From the Training Videos: Instead of using Web images, this baseline method employs frames randomly selected from the training videos to learn the model. With this baseline method, we aim to assess the benefit of introducing Web images for training.

C. Experimental Results

1) Action Recognition: The proposed approach is compared with the three baseline methods, and the results on the UCF sports dataset and the Olympic sports dataset are shown in Tables I and II, respectively. For fair comparison, we report the average results of ten iterations in our algorithm together with the standard deviation of the average. Three evaluation metrics (i.e., mean per class F-measure, overall accuracy and mean per class accuracy) are employed to compare our method with baseline methods. As shown in Tables I and II, the proposed approach significantly improves the recognition accuracy compared with the first two baseline methods in which no images are employed, and these results demonstrate the effectiveness of leveraging annotated images for training the model. Meanwhile, our method performs better than the third baseline method [i.e., TLSVM(video frames)], in which frames randomly selected from the training videos are employed. A major cause of the performance improvement is that our method avoids the problem of overfitting. Furthermore, the latent SVM method achieves better performance than the linear SVM method, which leads a conclusion

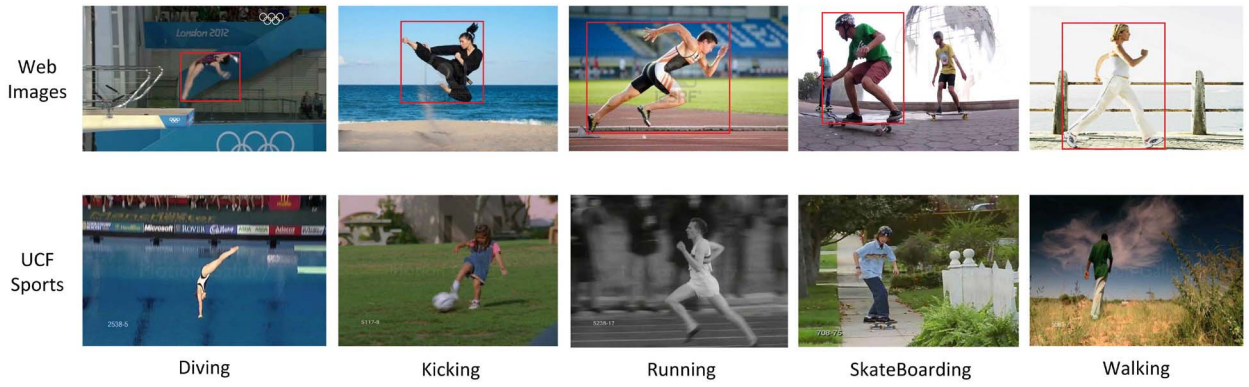


Fig. 5. Examples of the Web images and the UCF sports dataset.

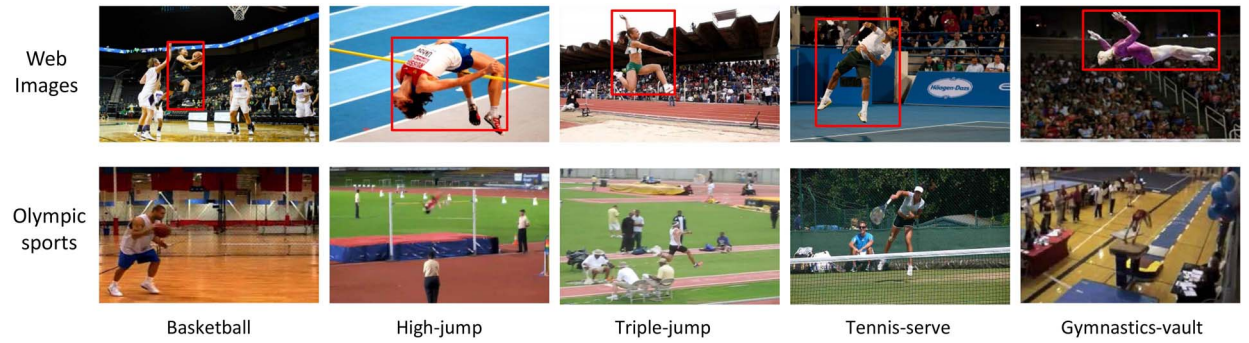


Fig. 6. Examples of the Web images and the Olympic sports dataset.

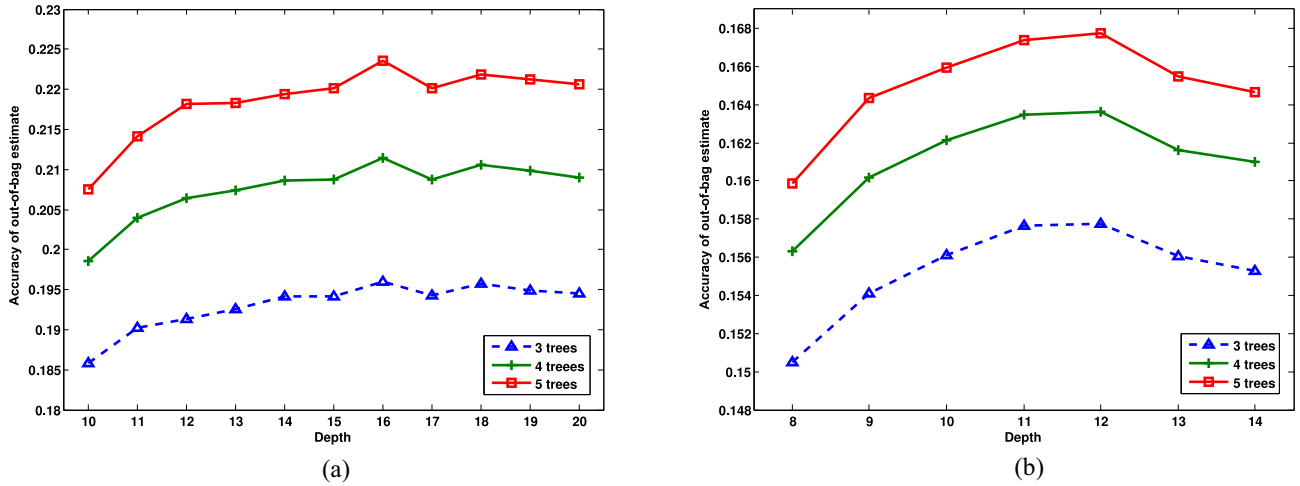


Fig. 7. Performance of clustering forests with different parameters. Performance of clustering forest on (a) videos and (b) Web images.

TABLE I
COMPARISON BETWEEN OUR METHOD AND THREE BASELINES ON THE UCF SPORTS DATASET

Method	mean per-class F-measure	overall accuracy	mean per-class accuracy
Linear SVM	0.704	0.681	0.711
Latent SVM	0.777(±0.0105)	0.760(±0.0099)	0.786(±0.0087)
TLSVM (Video frames)	0.789(±0.0251)	0.776(±0.0258)	0.800(±0.0228)
TLSVM (Web images)	0.817(±0.0275)	0.805(±0.0266)	0.826(±0.0244)

that incorporating local spatiotemporal information is able to benefit the recognition of action videos.

We also compare our method with state-of-the-art methods on the UCF sports dataset and the Olympic sports dataset. Here, we would like to emphasize that all the methods we compare to utilize the same test strategy as our method. Table III reports the mean per class recognition accuracy of different methods on the UCF sports dataset. Among these

TABLE II
COMPARISON BETWEEN OUR METHOD AND THREE BASELINES ON THE OLYMPIC SPORTS DATASET

Method	mean per-class F-measure	overall accuracy	mean per-class accuracy
Linear SVM	0.622	0.634	0.643
Latent SVM	0.651(±0.0049)	0.671(±0.0052)	0.673(±0.0049)
TLSVM (Video frames)	0.671(±0.0075)	0.678(±0.0052)	0.684(±0.0059)
TLSVM (Web images)	0.704(±0.0049)	0.712(±0.0039)	0.718(±0.0047)

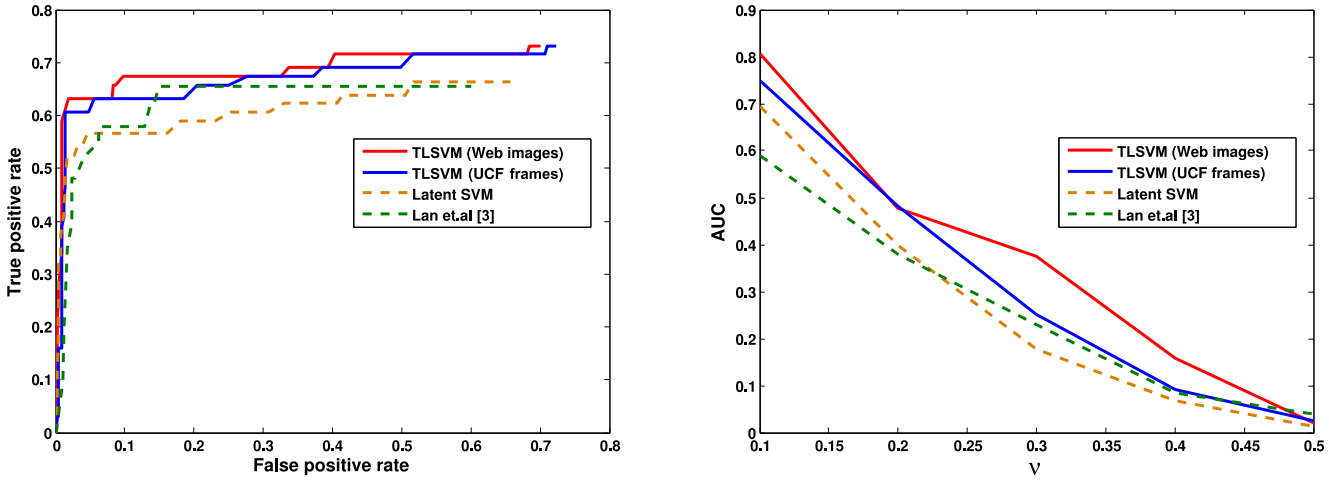


Fig. 8. Action localization performance comparison on the UCF sports dataset. (a) ROC curves for $\nu = 0.2$. (b) Area under ROC for different ν.

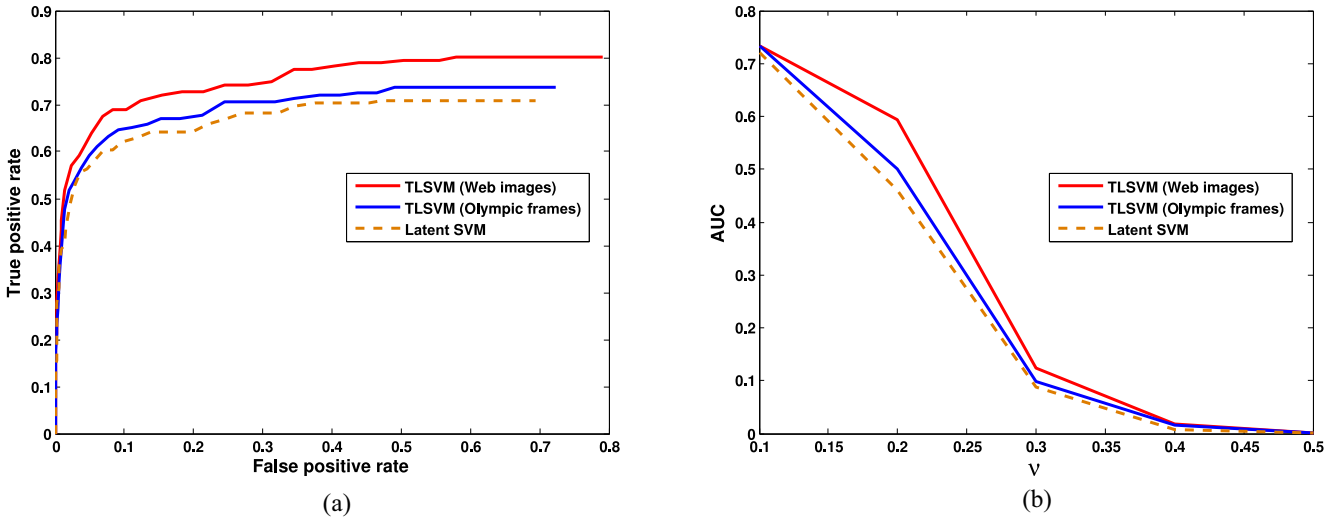


Fig. 9. Action localization performance comparison on the Olympic sports dataset. (a) ROC curves for $\nu = 0.2$. (b) Area under ROC for different ν.

methods, [5]–[7] are weakly supervised action recognition and localization methods, while [3] and [4] are supervised methods which require the hand-labeling of regions of actions in training videos. As shown in Table III, our method significantly outperforms three recently proposed weakly-supervised methods [5]–[7], which demonstrates the positive effect of considering the local similarities between spatiotemporal regions from training videos and annotated regions from Web images. It is also observed that our method performs better than [3] and [4], though our method does not require access to the ground-truth of action localizations in training videos. For comparison with state-of-the-art approaches on the Olympic sports dataset, we evaluate the mean average precision (MAP) for all categories and show the results of different methods in Table IV. From Table IV, we can observe that the proposed method is competitive with [37] and achieves better performance than [35] and [38]–[40]. One possible reason is that the introduction of local spatiotemporal regions of interest boosts the recognition of actions in videos. Besides, our method can simultaneously recognize and localize actions, while the compared methods in Table IV are designed only for action recognition.

Fig. 10. Visualization of action localization results of our method and [3] on the UCF sports dataset. Localization results of our method, [3], and the ground-truth are represented by red, purple, and green bounding boxes, respectively.

TABLE III
ACTION RECOGNITION ACCURACY COMPARISON WITH STATE-OF-THE-ART ON THE UCF SPORTS DATASET

Method	Accuracy
Lan et al. [3]	0.731
Shapovalova et al. [5]	0.753
Tian et al. [15]	0.752
Raptis et al. [4]	0.794
Ma et al. [6]	0.817
Boyraz et al. [7]	0.810
Our method	0.826 ($\pm$0.0244)

TABLE IV
MAP COMPARISON WITH STATE-OF-THE-ART METHODS ON THE OLYMPIC SPORTS DATASET

Method	MAP
Niebles et al. [35]	0.625
Tang et al. [38]	0.668
Liu et al. [39]	0.743
Li et al. [40]	0.765
Zhou et al. [37]	0.783
Our method	0.767 ($\pm$0.0014)

2) Action Localization: In order to evaluate the localization performance, we use the evaluation criterion in [3] and compute the receiver operating characteristic (ROC) curves for each action class. Given a video, the IOU score is computed for each frame, and the average IOU score over all test frames is compared to a predefined threshold ν to decide whether this video is successfully localized. A test video is considered to be correctly predicted if it is correctly classified and the average IOU score is larger than ν.

On the UCF sports dataset, we compare the proposed approach with the method of [3] as well as the last two baseline methods, and summarize the results in Fig. 8. On the Olympic sports dataset, the proposed method is compared with the last two baseline methods, and the comparison results are shown in Fig. 9. The average ROC curves with $\nu = 0.2$ are depicted in Figs. 8(a) and 9(a) for the UCF sports dataset and the Olympic sports dataset, respectively. The area under ROC curve is evaluated with ν varying from 0.1 to 0.5, and the curves are shown in Figs. 8(b) and 9(b).

From Figs. 8 and 9, it is observable that our method is able to achieve much better performance than the two baseline methods, which demonstrates the effectiveness of introducing the Web images for learning. Moreover, in many cases, the proposed approach using Web images performs better than the third baseline method which employs images from training data, especially for $\nu = 0.2$. These results demonstrate the positive effect of introducing Web images into training for action localization. As shown in Fig. 8, it is difficult for the "latent SVM" baseline to achieve comparable performance with [3], since [3] is trained on videos annotated with bounding boxes for each frame. However, our method is able to outperform [3] by using a few annotated images.

The action localization results of our method compared against [3] on the UCF sports dataset are shown in Fig. 10. Although we do not have access to ground-truth bounding boxes in training while [3] does, our method still achieves comparable or even better results on most videos. Fig. 11 visualizes the localization of actions on the Olympic sports dataset. As shown in Figs. 10 and 11, our method is able to correctly localize actions being conducted even in complex scenes, such as the videos of *Diving*, *Skateboarding*, *Hammer-throw*, and *Discus-throw*.

3) Computational Cost Analysis: We analyze the qualitative computational cost of the proposed method, and compare

Fig. 11. Visualization of action localization results of our method on the Olympic sports dataset. Localization results of our method and the ground-truth are represented by red and green bounding boxes, respectively.

it with [3] and [5]. The computational complexity for the inference algorithm defined in (12) is $O(N_Y N_H)$, where N_Y is the number of action classes and N_H is the number of spatiotemporal regions of interest. Generally, we generate tens of spatiotemporal regions for an input video. The computational complexity of [5] is also $O(N_Y N_H)$. In [3], the inference problem is solved iteratively, and the computational complexity is $N_Y N_F N_L$, where N_F is the number of frames of an input video and N_L is the number of bounding boxes in each frame. In their implementation, N_L is set to be 100, and N_F varies for different videos. Thus, the computational complexity of our method is lower than that of [3].

VII. Conclusion

We have presented a discriminative TLSVM for jointly recognizing and localizing actions in videos. The model is trained on videos only annotated with action labels, and a few Web images annotated with both action label and action location are introduced into the learning framework. The spatiotemporal region capturing the action being performed is treated as a latent variable in the proposed model, and an unsupervised method is utilized to extract a set of candidate spatiotemporal regions from a given video. Since images and videos are represented by different types of features, we introduce a structural transformation that maps images to videos. Experimental results on the UCF sports and Olympic sports datasets demonstrate that our model can effectively recognize and localize actions in videos.

References

[1] A. Oikonomopoulos, I. Patras, and M. Pantic, "An implicit spatiotemporal shape model for human activity localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Miami, FL, USA, 2009, pp. 27–33.

[2] A. Yao, J. Gall, and L. Van Gool, "A Hough transform-based voting framework for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 2061–2068.

[3] T. Lan, Y. Wang, and G. Mori, "Discriminative figure-centric models for joint action localization and recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 2003–2010.

[4] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1242–1249.

[5] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, and G. Mori, "Similarity constrained latent support vector machine: An application to weakly supervised action classification," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 55–68.

[6] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff, "Action recognition and localization by hierarchical space-time segments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 2744–2751.

[7] H. Boyraz, S. Z. Masood, B. Liu, M. Tappen, and H. Foroosh, "Action recognition by weakly-supervised discriminative region localization," in *Proc. British Mach. Vis. Conf.*, Nottingham, U.K., 2014.

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.

[9] A. Mansur, Y. Makihara, and Y. Yagi, "Inverse dynamics for action recognition," *IEEE Trans. Cybern.*, vol. 43, no. 4, pp. 1226–1236, Aug. 2013.

[10] X. Wu, D. Xu, L. Duan, J. Luo, and Y. Jia, "Action recognition using multilevel features and latent structural SVM," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1422–1431, Aug. 2013.

[11] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 810–822, Feb. 2014.

[12] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal Laplacian pyramid coding for action recognition," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 817–827, Jun. 2014.

[13] G. Luo *et al.*, "Learning human actions by combining global dynamics and local appearance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2466–2482, Dec. 2014.

[14] D. J. Cook, N. C. Krishnan, and P. Rashidi, "Activity discovery and activity recognition: A new partnership," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 820–828, Jun. 2013.

[15] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 2642–2649.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[17] L. Cao, Z. Liu, and T. S. Huang, "Cross-dataset action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 1998–2005.

[18] B. Z. Yao, B. X. Nie, Z. Liu, and S.-C. Zhu, "Animated pose templates for modeling and detecting human actions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 436–452, Mar. 2014.

[19] M. R. Amer and S. Todorovic, "A chains model for localizing participants of group activities in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 786–793.

[20] L. Duan, D. Xu, and S.-F. Chang, "Exploiting Web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1338–1345.

[21] L. Chen, L. Duan, and D. Xu, "Event recognition in videos by learning from heterogeneous Web sources," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 2666–2673.

[22] J. Chen, Y. Cui, G. Ye, D. Liu, and S.-F. Chang, "Event-driven semantic concept discovery by exploiting weakly tagged Internet images," in *Proc. ACM Int. Conf. Multimedia Retriev.*, Dallas, TX, USA, 2014, p. 1.

[23] N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff, "Learning actions from the Web," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, 2009, pp. 995–1002.

[24] N. Ikizler-Cinbis and S. Sclaroff, "Web-based classifiers for human action recognition," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1031–1045, Aug. 2012.

[25] F. Moosmann, E. Nowak, and F. Jurie, "Randomized clustering forests for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1632–1646, Sep. 2008.

[26] J. C. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, 2007, pp. 1–8.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. San Diego, CA, USA, 2005, pp. 886–893.

[28] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 3169–3176.

[29] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[30] M. Leordeanu, R. Sukthankar, and C. Sminchisescu, "Efficient closed-form solution to generalized boundary detection," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 516–529.

[31] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[32] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[33] T.-M.-T. Do and T. Artières, "Large margin training for hidden Markov models with partially observed states," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, 2009, pp. 265–272.

[34] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH: A spatio-temporal maximum average correlation height filter for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, 2008, pp. 1–8.

[35] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proc. Eur. Conf. Comput. Vis.*, Heraklion, Greece, 2010, pp. 392–405.

[36] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[37] Q. Zhou, G. Wang, K. Jia, and Q. Zhao, "Learning to share latent tasks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 2264–2271.

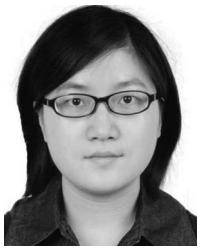
[38] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 1250–1257.

[39] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 3337–3344.

[40] W. Li and N. Vasconcelos, "Recognizing activities by attribute dynamics," in *Proc. NIPS*, Lake Tahoe, NV, USA, 2012, pp. 1115–1123.

Cuiwei Liu received the B.S. and Ph.D. degrees in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2009 and 2015, respectively.

She is currently a Lecturer with Shenyang Aerospace University, Shenyang, China. Her current research interests include computer vision, pattern recognition, and video understanding.



Xinxiao Wu received the B.S. degree in computer science and technology from the Nanjing University of Information Science and Technology, Nanjing, China, in 2005, and the Ph.D. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2010.

She is currently an Associate Professor with the Beijing Institute of Technology. Her current research interests include computer vision, machine learning, and video content analysis.

Yunde Jia (M'11) received the B.S., M.S., and Ph.D. degrees in mechatronics from the Beijing Institute of Technology (BIT), Beijing, China, in 1983, 1986, and 2000, respectively.

He is a Professor of Computer Science with BIT, and serves as the Director of the Beijing Laboratory of Intelligent Information Technology, Beijing. He has served as the Executive Dean of the School of Computer Science, BIT, from 2005 to 2008. He was a Visiting Scientist with Carnegie Mellon University, Pittsburgh, PA, USA, from 1995 to 1997, and a Visiting Fellow with the Australian National University, Canberra, ACT, Australia, in 2011. His current research interests include computer vision, media computing, and intelligent systems.