UCLA
UCLA Electronic Theses and Dissertations

Title
Mining Spatial and Spatio-Temporal ROIs for Action Recognition

Permalink
https://escholarship.org/uc/item/9gp7w2h3

Author
Lian, Xiaochen

Publication Date
2016

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California

Los Angeles

Mining Spatial and Spatio-Temporal ROIs for Action Recognition

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Statistics

by

Xiaochen Lian

2016

© Copyright by
Xiaochen Lian
2016

Abstract of the Thesis

Mining Spatial and Spatio-Temporal ROIs for Action Recognition

by

Xiaochen Lian

Master of Science in Statistics

University of California, Los Angeles, 2016

Professor Alan Loddon Yuille, Chair

In this paper, we propose an approach to classify action sequences. We observe that in action sequences the critical features for discriminating between actions occur only within sub-regions of the image. Hence deep network approaches will address the entire image are at a disadvantage. This motivates our strategy which uses static and spatio-temporal visual cues to isolate static and spatio-temporal regions of interest (ROIs). We then use weakly supervised learning to train deep network classifiers using the ROIs as input. More specifically, we combine multiple instance learning (MIL) with convolutional neural networks (CNNs) to select discriminative action cues. This yields classifiers for static images, using the static ROIs, as well as classifiers for short image sequences (16 frames), using spatio-temporal ROIs. Extensive experiments performed on the UCF101 and HMDB51 benchmarks show that both these types of classifiers perform well individually and achieve state of the art performance when combined together. We also show qualitatively that our ROIs (selected by the algorithms) capture the most relevant parts of the image sequences.

The thesis of Xiaochen Lian is approved.

Qing Zhou

Nicolas Christou

Alan Loddon Yuille, Committee Chair

University of California, Los Angeles

2016

To my mother and father ...

who forced me to learn program

while I was enjoying holidays after elementary school

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Recognition of human actions in realistic videos is a challenging problem [2], due to its complex content, cluttered background, and large intra-class variations caused by scale and location variations and viewpoint changes

Humans appear to tackle this challenge using two abilities: (i) The ability to rapidly detect static and spatio-temporal regions of interest (ROIs), instead of processing the entire image (e.g., bottom-up attention). (ii) The ability to determine which ROIs are useful for detecting specific actions and to extract the relevant visual cues for action discrimination. These ROIs contain the key information about the action. For example, static ROIs can include not only the people doing the action but also objects that people interact with (*e.g.* bicycles in *B*iking) or which often co-occur with the actions (*e.g.* basketboard in *B*asketball), and background context (*e.g.* swimming pool in swimming actions). Similarly, the spatio-temporal ROIs could be whole human body motion, the motion of body parts (*e.g.* hands in *P*ush Ups), movements of objects (*e.g.* barbell in *C*lean and Jerk), and even background motion (*e.g.* sea waves in *S*urfing).

These considerations motivate us to propose a video action recognition method that attends to regions of the videos, instead of the entire video. Fig. **??** illustrates the pipeline of our method, which consists of two models: the Static Model and the Motion Model. Both models mine ROIs in the video to obtain discriminative action cues: The Static Model takes image frames as input and uses low-level cues to propose static ROIs, e.g., 2D bounding boxes. More specifically, we use generic

object proposal methods [3, 4, 5]. The Motion Model works on video clips (*i.e.* a short sequence of frames), and mines spatio-temporal ROIs, which we call *v*ideo tubes. These are obtained using low-level cues followed by temporal grouping (details are given later).

Mining the ROIs is challenging because we do not know which ROIs are helpful for discriminating the actions. It would be helpful if the ROIs were annotated by action class, but this has only been done for humans (*e.g.* UCF101 [6], J-HMDB [7]). This means we cannot use fully supervised methods for mining the ROIs and must instead use weak supervision. More specially, we use multiple instance learning (MIL), where a video frame or a video clip is a "bag" and the ROIs are its "instances". We combine ML with deep convolutional neural networks (CNNs) to mine deep features from the ROIs. This enables both the Static and the Motion Models to classify image frames and video clips respectively. Our final system combines these classifiers.

The main contribution of this work is an action recognition model using spatial and spatio-temporal ROIs rather than the whole visual scene. In particular we improve upon existing methods in the following ways:

1. We generate ROIs to make proposals for video regions which contain discriminative cues for action recognition..

2. We formulate the ROI mining as an MIL problem and incorporate it into CNN structure, which enables unified learning of deep features and MIL.

3. Our model achieves state of the art performance on two action recognition datasets: UCF 101 [6] and HMDB51 [8]. We also show that the ROIs selected by the models capture the most relevant parts of the videos.

The rest of the thesis is organized as following: In Chapter **??** we briefly review the literature and related work. Then we describe the proposed method in

Chapter 3. Evaluation are given in Chapter 4 followed by conclusions and future work in Chapter 5.

CHAPTER 2

Related Work

?? Action recognition on videos has been extensively studied in computer vision community, and it is beyond the scope to review the entire literature. We refer readers to [2, 9] for a detailed survey .

Hand-crafted representations have been widely used, including low-level, mid-level and high-level features. Low-level ones extract representations [10, 11, 12] around interesting points [13] or trajectories [14, 15, 16] with Bag-of-Word (BOW) descriptors [17, 18, 19, 20, 21, 22]. Low-leve features suffer from their limited representation ability and discriminative capacity. To overcome this issue, several mid-level representations (*e.g.* Dynamic-Poselet [23], Motionlets [24], Actons [25] and [26]) and high-level representations (*e.g.* Action Bank [27]) have been proposed. The idea lies in discovering and mining representative visual/motion patterns or select discriminative elements in the action videos.

Recently, there have been attempts to learn deep representations for video action recognition [28, 29, 30, 31], motivated by the great success of deep learning techniques in image-based tasks [32, 33, 34, 35]. However, these deep models did not perform as well as the current best hand-crafted shallow representation [15]. The first deep learning framework with matching performance is Two-Stream network [36, 1], which uses two separate CNNs to model color and motion with a final fusion. Different from their model where features are extracted on the whole spatial extent, our model utilizes local regions (in the Static Model) and flexible video tubes (in the Motion Model). Wang *et al.* [37] applied the trajectory-based

pooling on the convolutional descriptors output by the Two-Stream network and encoded them using Fisher vector. Their pooling strategy shares some similarities with our motion tubes. CNNs with 3D convolution operations [38, 39] have been proposed to preserve the temporal information of the input signals and enable shift-invariance in the temporal domain.

Deep representations to encode long-term temporal structure have also been attracted some attention [40, 41, 42, 43]. Recurrent Neural Networks have been used to model a sequence of transformation across frames [44, 45, 46, 47, 48, 49, 50]. Our model processes videos at the level of short clips and can be readily plugged into these systems as a feature representation component.

The proposals in our method can be interpreted as "parts" (of videos), the concept which is originated from image understanding [51, 52], and has been introduced to video classification [24, 25, 26]. Our method is different from these methods in that it incorporates the deep representation of "parts" (*i.e.* the proposals) in the model and provides a joint learning framework, as in [53] but with a "deeper" network architecture.

Visual attention have been largely explored in image and video classification tasks [54, 55], either implicitly [56, 57]. or explicitly [58, 59, 60, 61]. Sharma *et al.* [57] adopted the implicit way to action recognition, by ensembling features from different spatial/temporal structure by a soft weighted voting. It turned out to be not the best choice in terms of performance. Our proposal based method [62, 63, 64] adopt the explicit strategy, by generating spatial and spatio-temporal proposals as candidate action-related elements.

Multiple instance learning (MIL) has been largely used to combine proposals in computer vision tasks [65, 66, 67, 68, 25]. It has been recently unified within deep learning frameworks [69, 70, 71, 64, 50]. The one most related to our work is [64], where some regions in the proximity of the query region of an image are were chosen as instances. In this paper, we do not have query regions nor corresponding

annotations. We extend their effort to video data by considering all possibly useful spatial-temporal constituents of videos.

CHAPTER 3

Approach

In this section, we describe in details the Static Model and the Motion Model. Both models have three components: ROI proposal generation, computation of deep features within ROIs, and training the deep network using MIL (after encoding and aggregation of the ROI deep features).

The ROI proposal algorithms are low-level and class-agnostic, since learning proposals would require annotated ROIs. We use an ROI ranking mechanism, so that our models only need to process a few, top scored, ROIs. This saves computation time and simplifies learning discriminative classifiers. Deep convolutional features are computed for ROIs, which are then encoded and aggregated using MIL.

3.1 Static Model

Fig. 3.1 shows the pipeline of the Static Model. Given an image frame I from a video, the first step is generating a set of candidate regions, which will be used as instances in the MIL framework. We use video labels as bag labels and the labels of instances (*i.e.* of the ROIs) are unknown and treated as latent variables. The deep convolutional features of the candidate regions are instance-level features. Next, the MIL component of the Static Model encodes the instance features, and learns the action classification model using the video class label.

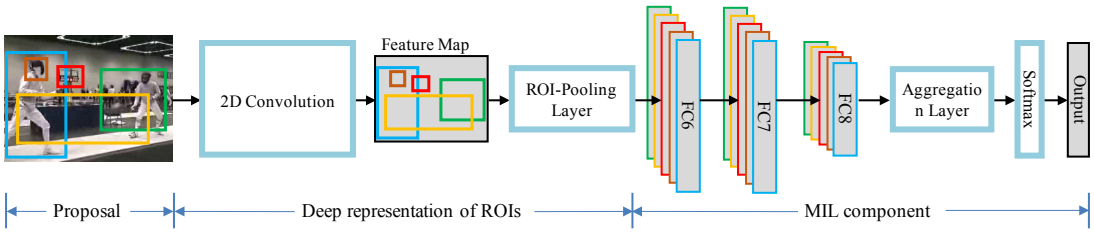

Figure 3.1: The network architecture of the proposed Static Model. Given an image frame I, a set of 2D bounding boxes (indicated by colors) are selected as candidate ROIs. The deep convolutional feature map of I are computed and pooled over each ROI. The pooled features are then passed to the MIL component, which is composed of three fully connected layers encoding features, an aggregation layer mapping encoded instance features into one bag-level feature, and a softmax layer that transforms the learned bag-level feature into final scores of actions.

3.1.1 Spatial ROI Proposals

To obtain a list of K regions of interest (ROIs) $R(I) = \{r_1, \ldots, r_K\}$ from frame I, we use the formulation of Edge Boxes [5][1], which estimates estimates bounding boxes for objects based on the amount of contours wholly within the box, together with an "objectiveness" score. After obtaining ROIs from Edge Boxes, we remove small boxes (*i.e.* with shorter side less than 50 pixels), and keep K boxes with highest "objectiveness" scores. We also include the whole frame region in case the full background context is needed. Fig. 3.2 shows some examples. In Section 4, we will see that with K as small as 20 our model can achieve very good performance.

3.1.2 Deep Instance Features

For each ROI r_k in $R(I)$, we compute the deep instance features $f(r_k, I; w_f)$ within it where w_f are the parameters of the CNN (initially pre-trained and then

[1]In fact, any object proposal methods that has a ranking mechanism among proposals can be used in our model, *e.g.* [4].

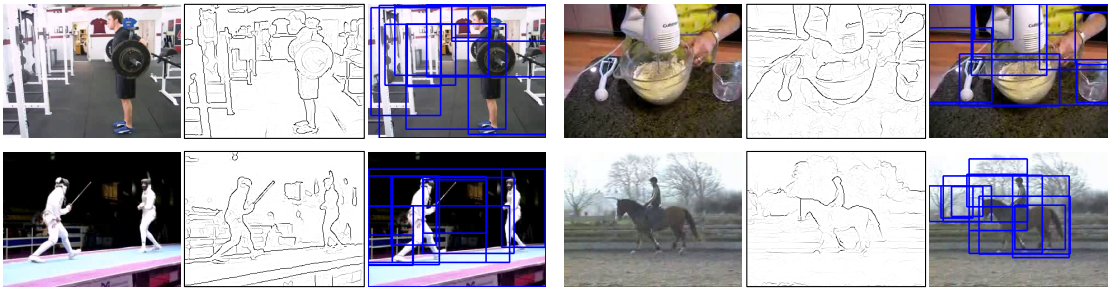

Figure 3.2: Four examples of our region proposals for Static Model. For each example, the left is the original frame image, the middle is the edge map, the right shows top 10 bounding box ROIs.

learnt by MIL). To compute the features efficiently, we perform convolutions at the frame level, and feed the convolutional feature map and $R(I)$ into the ROI Pooling layer [72]. This converts the features inside r_k into a feature map with a fixed spatial extent of H×W (*e.g.* 7×7 in our experiments).

3.1.3 Multiple Instance Learning

The instance features of the regions in $R(I)$ are then passed to the MIL component shown in Fig. 3.1, which has three steps: First, the instance features are encoded through three fully connected layers $FC6$, $FC7$ and $FC8$, which is formulated as

$$s_k = e(f(r_k, I; w_f); w_e), \quad k = 1, \ldots, K \tag{3.1}$$

where e represents the encoding with parameters w_e, $s_k \in \mathbb{R}^D$ is the encoded features. Second, $\{s_k\}_{k=1}^K$ are mapped to one bag-level feature by the aggregation function g:

$$h(W) = g(s_1, \ldots, s_k; w_g) \tag{3.2}$$

where w_g is the parameters of $g(\cdot)$, $h \in \mathbb{R}^C$ is the bag-level feature, $W = \{w_f, w_e, w_g\}$ is the parameters of the whole network, and C is the number of classes. The aggregation function $g(\cdot; w_g)$ can be any function that maps multiple features into one feature and that can be blended into the gradient descent

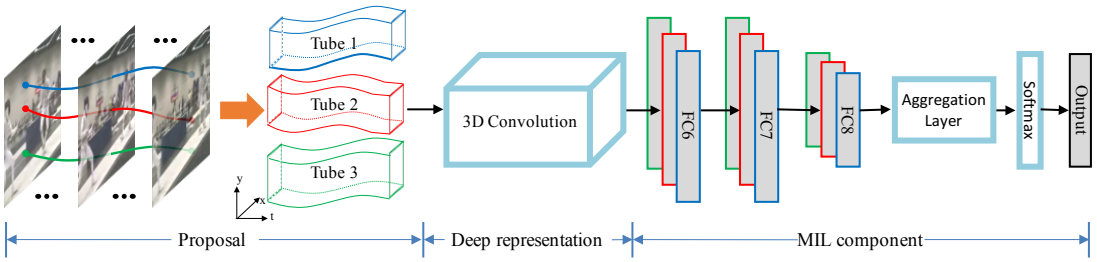

Figure 3.3: The network architecture of the proposed Motion Model. Given a video clip, a set of video tubes (indicated by colors) are selected as candidates spatio-temporal ROIs. A 3D Convolutional network is used to compute the features of these tubes and the outputs are input into the MIL component, which is composed of three fully connected layers encoding features, an aggregation layer mapping instance-level features into one bag-level feature, and a softmax layer that transforms the learned bag-level feature into final scores of actions.

optimization mechanism. In this paper, we compare two simple functions: max pooling and average pooling, and therefore the dimension of s_k is equal to C. Thirdly, the bag feature h is transformed into the action scores of C classes,

$$p_c = \frac{\exp(h_c)}{\sum_{i=1}^{C} \exp(h_i)}, \quad c = 1, \ldots, C \tag{3.3}$$

and the loss we use is the cross-entropy classification loss *i.e.* $-\log(p_{\hat{c}})$ where $\hat{c}$ is the ground truth class label of I.

3.2 Motion Model

Fig. 3.3 shows the pipeline of our Motion Model, which is composed of low-level proposals to obtain spatio-temporal ROIs (video tubes for short) followed by the multiple instance learning of action classification. The basis structure is almost identical to the Static Model.

The objective of the local motion proposal step is to generate a set of spatio-temporal ROIs of videos, which may contain cues for the action. Unlike previous

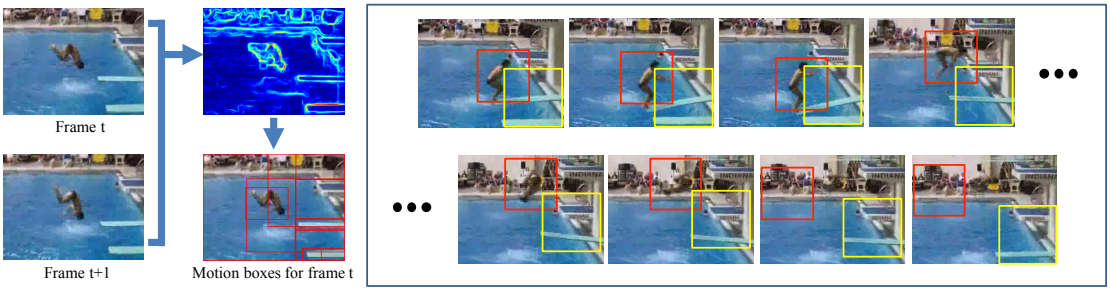

Figure 3.4: Left: Motion box generation on a single frame: two consecutive frames are used to estimation the motion boundaries which is then used as edge map input for Edge Boxes to produce motion boxes (red bounding boxes). Right: Two video tubes proposals on the first four and last four frames of a 16-frame video clips. Boxes with same color belong to the same video tube. The red tubes localizes the diver and the yellow one finds the diving board.

works on action detection [63, 73, 74] and action proposals [75] which focus only on human motions, we also consider the movements of objects and even some backgrounds. These types of background movement are often very useful to help identify actions (*e.g.* the motion of a road as a biker cycles down it). We propose an approach for video tube generation which provides tight spatio-temporal localization of the motions in videos. Afterwards, we formulate the learning of deep representation of video tubes and MIL component in a unified network enabling joint learning.

3.2.1 Video Tubes

Given a video clip of L frames $V = (I^1, \ldots, I^L)$, the goal of this step is to propose a set of K spatio-temporal ROIs, or video tubes $T = \{t_1, \ldots, t_K\}$, where each tube $t_k = (r_k^1, \ldots, r_k^L)$ is a temporal series of 2D bounding boxes that localize motions. We call these 2D bounding boxes "motion boxes". Our algorithm build up a video tube from a single image frame, by generating motion boxes on individual image frames and then linking the boxes across frames to form video tubes.

The left part of Fig. 3.4 illustrates motion box generation on a single frame I. Unlike the object boxes in the Static Model, motion boxes are intended to capture moving parts in the video. We apply Edge Boxes again, but use the motion boundaries [76] detected based on two consecutive image frames as edge map. In this case, the objectiveness score estimated by Edge Boxes actually reflects the amount of motion contours within in a motion box b, which we call the "motionness" score $m(b)$.

Once we have motion boxes on individual frames, we produce a set of video tubes by linking boxes across frames. A good video tube proposal t_k should have a high motionness score, *i.e.* $m(t_k) = \sum_{l=1}^{L} m(r_k^l)$ is large, and have spatio-temporal smoothness:

$$\text{IOU}(r_k^l, r_k^{l+1}) \geq \sigma_o, \quad l = 1, \ldots, L-1 \tag{3.4}$$

and have consistent appearance along the tube, *i.e.*

$$\parallel A(r_k^l) - A(r_k^{l+1}) \parallel_2 \leq \sigma_a, \quad l = 1, \ldots, L-1 \tag{3.5}$$

where σ_o and σ_a are thresholds, $A(\cdot)$ compute the color histogram within a box. In this paper, we use $\sigma_o = 0.5$, $\sigma_a = 0.2$ and divide R, G and B channels into 16 bins when computing color histogram.

Now for each motion box b_i^L in the last frame I^L of V, we compute the best tube ending at b_i^L, using dynamic programming,

$$f(b_i^l) = \max_{b_j^{l-1} \in I^{l-1}} f(b_j^{l-1}) + m(b_i^l) + d(b_i^l, b_j^{l-1}) \tag{3.6}$$

where $d(b_i^l, b_j^{l-1})$ is $-\infty$ if b_i^l and b_i^{l-1} do not satisfy the constraints in Eq. 3.4 and Eq. 3.5, and is equal to 0 otherwise. Then we can back-trace from every $b_i^L \in M(I^L)$ to recover a video tube. This yields a large amount of tubes. Finally, we apply non-maximum suppress to prune out highly overlapping video tubes, according to their motionness scores.

For each remaining video tube, say t_k, we first crop from l-th frame a square patch p_k^l with its center at the center of r_k^l and size

$$a = \max\left(\text{median}(h(r_k^1), \ldots, h(r_k^l)), \text{median}(w(r_k^1), \ldots, w(r_k^l)\right) \tag{3.7}$$

where $h(\cdot)$ and $w(\cdot)$ returns the height and the width of a bounding box respectively. We then update t_k by replacing r_k^l with p_k^l and obtain the final video tube t_k. The right part of Fig. 3.4 shows two example video tubes.

3.2.2 Deep Instance Features

There are several options for computing the deep features of a video clip, *e.g.* [31, 36, 38, 44, 45]. We choose the 3D convolutional network (C3D) in [38], due to its good performance and the convenience of joint end-to-end training. In C3D, traditional 2D convolution and 2D pooling operations are replaced with the 3D version, *i.e.* with an additional temporal dimension, to prevent the temporal information from being collapsed. We use the output of the last convolution layer as the instance feature, whose temporal dimension of the outputs becomes 1.

3.2.3 Multiple Instance Learning

The instance features in V are passed through the MIL component in Fig. 3.3, which is essentially the same as the MIL component in the Static Model: the instance features are encoded through three fully connected layers, and then are mapped to one bag-level feature, which is transformed into the action scores of C classes. The parameters of the network are trained using cross-entropy loss.

CHAPTER 4

Experiments

In this section, we first introduce the details of our experimental settings. Then we provide quantitative and qualitative results.

4.1 Datasets

The evaluation is performed on UCF101 [6] and HMDB51 [8] benchmarks. UCF101 contains $13,320$ videos of 101 action classes; HMDB51 includes $6,766$ videos of 51 actions. In both datasets, the videos of the same action class are grouped into several groups; The videos from the same group may share some common features, such as similar background, similar viewpoint, *etc.* . Both datasets provide three official splits into training and test data. All the partitions satisfy that the videos belonging to the same group are kept separated in training and testing. The performance is measure by the average classification accuracy across the splits.

We begin by conducting diagnostic experiments on the first split of UCF101 dataset (UCF split1). For comparison with the state of the art, we follow the standard evaluation protocol on both UCF101 and HMDB51.

4.2 Implementation Details

4.2.1 Static Model

For UCF101 dataset, we initialize the parameters of the Static Model (*i.e.* 13 convolutional layers and the first two feature encoding layers) with VGG-16 [33] pre-trained on ImageNet dataset. The last encoding layer FC8 is initizlied with random weights. For HMDB51 dataset, as the number of training videos are relativel small (around 3.7K), we fine-tune the Static Model tarined on all videos of UCF101 datasets. The learning rate starts with 0.001, decreases to its 1/10 every $4,000$ iterations and stops at $10,000$ iterations. The dropout ratios for the encoding layers are set to be 0.5, as we observed performance degradation with higher dropout ratios. The corner cropping and the multi-scale cropping suggested in [1] are used on video frames of size 256×340 to get cropped frames of size 224×224, which are later horizonatally flipped with probability 50%. The ROIs in the images are transformed accordingly. At the test time, we sample 25 frame images. From each of these selected frames, we obtain 10 regions, i.e. 4 corners, 1 center, and their horizontal flippings. The final prediction score is obtained by averaging across the sampled frames and their cropped regions.

4.2.2 Motion Model

For UCF101 dataset, we initialize the parameters of the Motion Model (*i.e.* 8 3D convolutional layers and the first two feature encoding layers) with C3D [38] pre-trained on Sports-1M dataset [31]. The last encoding layer FC8 is initialized with random weights. For HMDB51 dataset, we again fine-tune the Motion Model trained on UCF101 datasets due to the smaller size of the dataset. The learning rate starts with 0.0001, decreases to its 1/10 every $10,000$ iterations and stops at $20,000$ iterations. The dropout ratios for the encoding layers are set to be 0.5. We use horizontal flipping as data augmentation. The videos are split into non-

Table 4.1: Average accuracy of the variants of the proposed Static Model on UCF split1.

	max	avg
ROIs=5	76.6%	76.3%
ROIs=10	78.3%	78.0%
ROIs=20	**81.0**%	80.4%
ROIs=40	79.2%	77.1%

overlapped 16-frame clips. For each clip, the video tube proposals are generated and resized to 112×112 (the input size used in [38]), and then are used as the input to the Model Motion. At the test time, 10 clips of 16 frame long are sampled, and the final prediction score is obtained by averaging across all the clips.

4.2.3 Model Fusion

We perform the inference with the two models separately. For each video, we use a weighted linear combination of the prediction scores produced by the two models. Note that there is no official way of tuning hyper-parameters (fusion weights in our case) on either UCF101 or HMDB51. We randomly choose two groups of videos from the training partition of UCF101 split1 as validation set and repeat the process three times. The weight is determined as 1 for the Static Model and 2 for the Motion Model.

In Section 4.4, we also combine our model (Static Model+Motion Model) with the Two-Stream Model in [1]. We simply put equal weights on the scores.

4.3 Diagnostic Experiments

All the experiments in this sub-section are conducted on UCF split1.

Table 4.2: Comparison between the Static Model and baselines on UCF101 split1.

methods	avg. accuracy
SPNet-CC-avg	79.8%
SPNet-CC-max	79.8%
SPNet-FC-avg	79.9%
SPNet-FC-max	79.6%
SPNet-ROI(20)-max	75.0%
S-ROI(20)-max	**81.0**%

4.3.1 Static Model

We first experiment with two aggregation functions, max and avg, and different number of ROIs (*i.e.* 5, 10, 20 and 40). The results are shown in Table 4.1. We can see that in all cases, max(·) performs better than avg. Using 5 or 10 ROIs per frame performs slightly worse than using 20 ROIs. The reason may be that fewer ROIs are not enough to cover all useful regions. However, using more ROIs does not necessarily bring better results, as more unrelated regions are brought in and we need more training data to handle that. In the following experiments, we will use max as the aggregation function and 20 ROIs per frame, which is denoted by I-max-ROI(20).

Next, we compare I-max-ROI(20) with the spatial net trained in [1], denoted by SPNet. In the paper, the final prediction score of a frame is obtained by averaging across the 10 regions generated by corner cropping (CC) and horizontal flipping. We call this approach SPNet-CC-avg. An alternative is doing max pooling across the 10 regions, *i.e.* SPNet-CC-max. We experiment with another two baselines using the spatial net of [1]: SPNet-FC-max and SPNet-ROI(20)-max. For SPNet-FC-max, we use the spatial net in the fully convolutional (FC) manner [77] on the whole video frame (of size 256×340) and apply max pooling across locations of the final score map of size 2×5, which can be considered

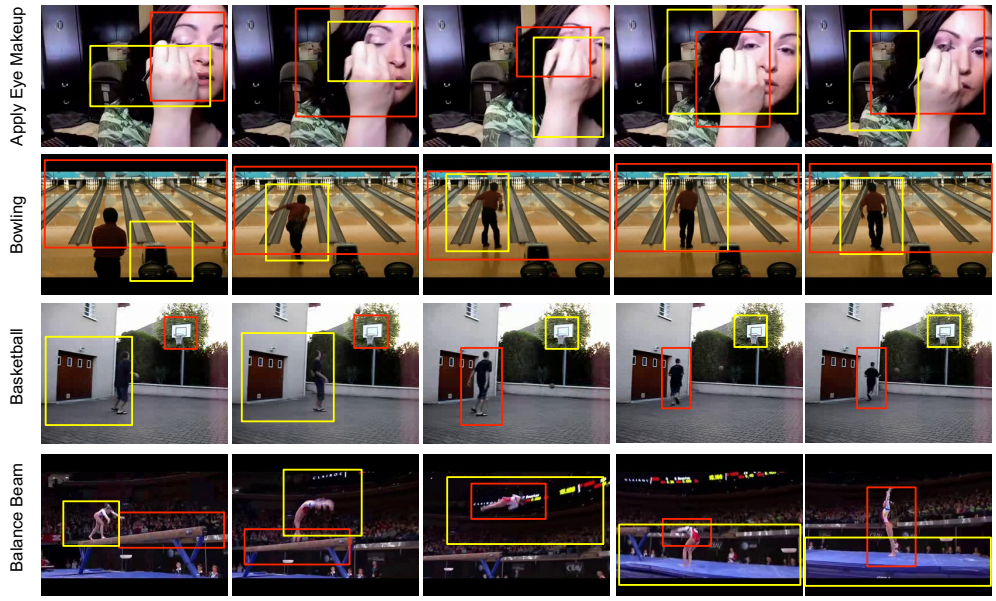

Figure 4.1: Visualization of the top two regions selected by S-ROI(20)-max. Each row corresponds to a video from the test partition of UCF101 split1. Red box corresponds to the top score one, and the yellow is the second best one. For each video we display five frames with equal temporal intervals.

as using dense and fixed-size proposals. The last one is SPNet-ROI(20)-max, where we combine SPNet with our proposal strategy. From the results shown in Table 4.2, we can see that S-ROI(20)-max outperforms SPNet-CC-{avg,max} and SPNet-FC-max, which proves that our proposal strategy is better than the fixed and the dense region proposal strategies. S-ROI(20)-max also performs better than SPNet-ROI(20)-max, which indicates the importance of the unified learning in our model.

In Figure 4.1, we show qualitatively that the Static Model can capture the relevant parts of the video, by visualizing the top scored spatial ROIs selected by S-ROI(20)-max, using videos from the test partition of UCF101 split1. From the figure we can see that the Static Model is able to find objects related to actions.

Table 4.3: Average accuracy of the variants of the proposed Motion Model on UCF split1.

	max	avg
ROIs=5	81.8%	81.7%
ROIs=10	**84.4**%	84.1%
ROIs=20	84.2%	84.0%

4.3.2 Motion Model

We first experiment with two aggregation functions, max and avg, and different number of video tube proposals (*i.e.* 5, 10 and 20). For convenience, we also call the proposed video tubes ROIs. The results are shown in Table 4.3. We can see that max(·) performs better than avg consistently. Using 5 ROIs per video clip performs worse than using 10 ROIs, which may be also due to the lack of enough ROIs to cover all useful regions. Using 20 ROIs does not give better performance either but demands more computational time. In the following experiments, we will use max as the aggregation function and 10 ROIs per video clip, which is denoted by M-max-ROI(10).

Next, we compare M-ROI(10)-max with the C3D network in [38], which is fine-tuned on UCF101 split1 from a C3D network pre-trained on Sports-1M. In [38], C3D used a single center crop per clip to make the prediction. Predictions of 10 clips randomly extracted from the video are averaged to give the video prediciton. We call this approach C3D-C_rand10. We also experiment with the corner croping (CC) and the fully convolutional (FC) schemes paired with max and average poolings, as we did for SPNet. These approaches can be seen as using 3D bounding boxes at fixed locations as proposals. The results are shown in Table 4.4, from which We can see that M-ROI(10)-max outperforms them, showing the advantage of adopting video tubes that track motions as proposals. We also combine C3D with our proposal strategy, *i.e.* C3D-ROI(10)-max. Without unified

Table 4.4: Comparison between the Motion Model and baselines on UCF101 split1.

methods	avg. accuracy
C3D-C_rand10	81.9%
C3D-CC-avg	82.2%
C3D-CC-max	82.1%
C3D-FC-avg	82.1%
C3D-FC-max	82.1%
C3D-ROI(10)-max	77.5%
M-ROI(10)-max	**84.4**%

learning of feature representation and MIL encoding, the performance drops.

In Figure 4.2, we visualize the top two scored spatio-temporal ROIs selected by M-ROI(10)-max, from which we can see that the Motion Model is able to find action-related spatio-temporal ROIs.

4.4 Comparison with The State of The Art

In this section, we first evaluate the Static Model and the Motion Model on all three splits of UCF101 and HMDB51. Then we compare with the state of the arts results. We also evaluate the fusion of the two models. Please refer to Section 4.2 for how to determine the fusion weights.

Table 4.5 and Table 4.6 show comparison between our models and the Two-Stream model in [1] on UCF101 and HMDB51. [1] used VGG-16 network [33] to boost the performance of the original Two-Stream model [36]. Note that [1] did not report experiments on HMDB51. We fine-tune the Two-Stream model pre-trained on UCF101, and denote this model as "Two-Stream by us". Our Static Model outperforms the spatial net (*i.e.* the network operating on individual frames) of

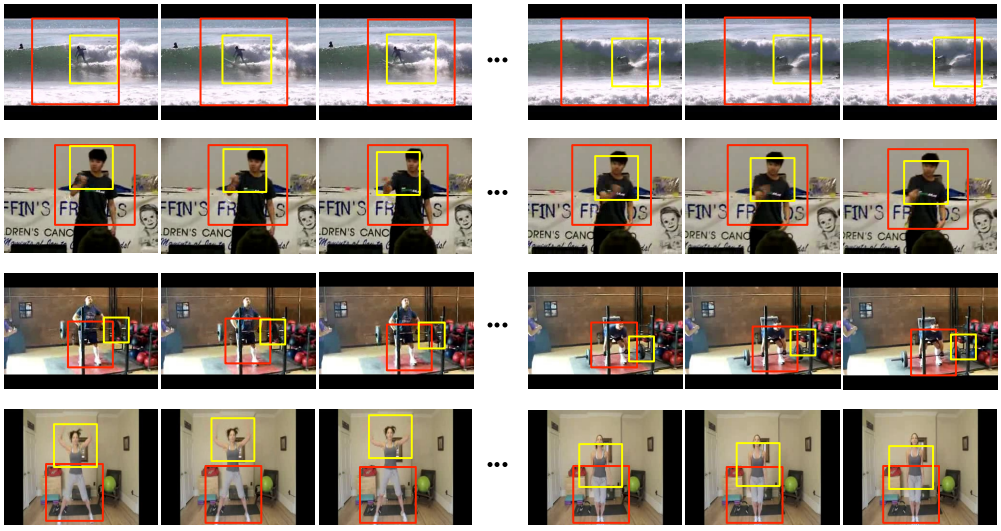

Figure 4.2: Visualization of the top two scored regions selected by M-ROI(10)-max. Each row corresponds to a video clip from the test partition of UCF101 split1. For each video clip we display first three and last three frames and omit the between. The red boxes correspond to the video tube with best action score, and the yellow is the one with second best score.

[1] in the static stream. In the motion stream, our Motion Model performs worse than the temporal net on UCF101. We argue that the temporal net uses 5 more convolution layers than us, we expect the Motion Model to get better results when fine-tuning from a deeper CNN. While on HMDB51, our Motion Model is better than the temporal net. The reason maybe HMDB51 has less training data; By attending to ROIs, our model suffers less from over fitting problem.

Table 4.7 presents action recognition accuracy of our method compared with current best methods. On UCF101, our method (Static Model + Motion Model) does not perform as well as [43, 1, 46]. However, when fused with the Two-Stream model [1], our method got a 2.3% performance gain and achieve the best performance. This shows that our models and the Two-Stream model are complementary to each other. In fact, the temporal net of Two-Stream uses stacked optical flow displacement fields as the input, while our Motion Model directly operates on RGB domain. On HMDB51, our method got the state of the art result on its own, and when combined with the Two-Stream model, the accuracy increases 2.2%.

Table 4.5: Comparison to the Two-Stream model [1].

UCF101		Two-Stream [1]	Ours
Static	split 1	79.8%	**81.0**%
	split 2	77.3%	**78.4**%
	split 3	77.8%	**78.8**%
	average	78.4%	**79.4**%
Motion	split 1	**85.7**%	84.4%
	split 2	**88.2**%	87.7%
	split 3	**87.4**%	86.5%
	average	**87.0**%	86.2%
Fusion	split 1	**90.9**%	89.8%
	split 2	**91.6**%	91.3%
	split 3	**91.6**%	90.3%
	average	**91.4**%	90.5%

Table 4.6: Comparison to the Two-Stream model from us.

HMDB51		Two-Stream by us	Ours
Static	split 1	54.3%	**57.0**%
	split 2	50.3%	**52.6**%
	split 3	50.1%	**52.6**%
	average	51.6%	**53.9**%
Motion	split 1	65.6%	**66.8**%
	split 2	62.4%	**64.3**%
	split 3	62.0%	**64.0**%
	average	63.3%	**65.0**%
Fusion	split 1	70.1%	**72.0**%
	split 2	67.2%	**68.2**%
	split 3	66.8%	**68.4**%
	average	68.0%	**69.5**%

Table 4.7: Comparison with the state of the art results.

HMDB51		UCF101	
IDT+FV [15]	57.2%	IDT+FV [15]	85.9%
Two-Stream [36]	59.4%	Hybrid [20]	87.9%
H-VLAD [22]	59.8%	Two-Stream [36]	88.0%
Hybrid [20]	61.1%	LSTM+Two-Stream [44]	88.6%
TDD+FV [37]	63.2%	C3D+iDT+SVM [38]	90.4%
Two Stream Siamese [43]	63.4%	Hybrid LSTM [46]	91.3%
SFV [21]	66.8%	Two Stream [1]	91.4%
Two-Stream by us	68.4%	Two-Stream Siamese [43]	92.4%
Ours	**69.5**%	Ours	90.5%
Ours+Two-Stream by us	**71.7**%	Ours+Two-Stream [1]	**92.8**%

CHAPTER 5

Conclusion and Future Work

In this work, we introduce a novel deep action recognition method with ROIs. By exploiting video benchmarks, we find that critical representations occur with in sub-regions of videos. Based on this observation, we extract static and spatio-temporal regions of interest (ROI) to enhance the performance of deep network. Features from different instances are naturally integrated into our MIL framework to adaptively select the most discriminative ROIs to enable end-to-end learning. Extensive experiments on UCF 101 and HMDB51 benchmarks demonstrate that our algorithm not only outperform existing methods quantitatively, but also capture the most relevant part qualitatively.

In the future, we will try deeper network structure for further improvement. Also, to exploit more aggregation functions in MIL other than average and max pooling is also desirable. Furthermore, to construct detailed annotation on the benchmark, i.e., high-quality hand-labeled regions, may be helpful to study the influence of different ROI generation schemes. One straightforward future work is to make our Motion Model deeper, *i.e.* using VGG-16 network as the starting point. The second possible future work is to exploit more aggregation functions for our MIL components. Another direction is constructing detailed annotation on widely used action recognition benchmark, *i.e.* annotating action-related regions and motions, with which we can explicitly model the proposals selection, and which motivates the study of action detection to consider non-human regions.

References

[1] Wang, L., Xiong, Y., Wang, Z., Qiao, Y.: Towards Good Practices for Very Deep Two-Stream ConvNets. ArXiv e-prints (July 2015)

[2] Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. ACM Computing Surveys **43**(3) (2011) 16

[3] Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**(2) (2013) 154–171

[4] Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: Bing: Binarized normed gradients for objectness estimation at 300fps. In: CVPR. (2014) 3286–3293

[5] Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV. Springer (2014) 391–405

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

[7] Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV. (December 2013) 3192–3199

[8] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV, IEEE (2011) 2556–2563

[9] Poppe, R.: A survey on vision-based human action recognition. Image and vision computing **28**(6) (2010) 976–990

[10] Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR, IEEE (2008) 1–8

[11] Klaser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC, British Machine Vision Association (2008) 275–1

[12] Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: ECCV. Springer (2006) 428–441

[13] Laptev, I.: On space-time interest points. IJCV **64**(2-3) (2005) 107–123

[14] Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR, IEEE (2011) 3169–3176

[15] Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV. (2013) 3551–3558

[16] Lan, Z., Lin, M., Li, X., Hauptmann, A.G., Raj, B.: Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In: CVPR. (2015) 204–212

[17] Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: CVPR, IEEE (2007) 1–8

[18] Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: ECCV. Springer (2010) 143–156

[19] Jain, M., Jégou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR. (2013) 2555–2562

[20] Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. arXiv preprint arXiv:1405.4506 (2014)

[21] Peng, X., Zou, C., Qiao, Y., Peng, Q.: Action recognition with stacked fisher vectors. In: ECCV. Springer (2014) 581–595

[22] Peng, X., Wang, L., Qiao, Y., Peng, Q.: Boosting vlad with supervised dictionary learning and high-order statistics. In: ECCV. Springer (2014) 660–674

[23] Wang, L., Qiao, Y., Tang, X.: Video action detection with relational dynamic-poselets. In: ECCV. Springer (2014) 565–580

[24] Wang, L., Qiao, Y., Tang, X.: Motionlets: Mid-level 3d parts for human motion recognition. In: CVPR. (2013) 2674–2681

[25] Zhu, J., Wang, B., Yang, X., Zhang, W., Tu, Z.: Action recognition with actons. In: ICCV. (2013) 3559–3566

[26] Jain, A., Gupta, A., Rodriguez, M., Davis, L.: Representing videos using mid-level discriminative patches. In: CVPR. (2013) 2571–2578

[27] Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1234–1241

[28] Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: ECCV. Springer (2010) 140–153

[29] Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR, IEEE (2011) 3361–3368

[30] Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. TPAMI **35**(1) (2013) 221–231

[31] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR. (2014) 1725–1732

[32] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1097–1105

[33] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015)

[34] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015) 1–9

[35] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)

[36] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS. (2014) 568–576

[37] Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: CVPR. (2015) 4305–4314

[38] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. ICCV (2014)

[39] Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. (2015)

[40] Tang, K., Fei-Fei, L., Koller, D.: Learning latent temporal structure for complex event detection. In: CVPR, IEEE (2012) 1250–1257

[41] Fernando, B., Gavves, E., Oramas, J.M., Ghodrati, A., Tuytelaars, T.: Modeling video evolution for action recognition. In: CVPR. (2015) 5378–5387

[42] Miao, J., Xu, X., Qiu, S., Qing, C., Tao, D.: Temporal variance analysis for action recognition. TIP **24**(12) (2015) 5904–5915

[43] Wang, X., Farhadi, A., Gupta, A.: Actions˜ transformations. arXiv preprint arXiv:1512.00795 (2015)

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

[45] Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In Blei, D., Bach, F., eds.: ICML, JMLR Workshop and Conference Proceedings (2015) 843–852

[46] Wu, Z., Wang, X., Jiang, Y.G., Ye, H., Xue, X.: Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, ACM (2015) 461–470

[47] Sun, C., Shetty, S., Sukthankar, R., Nevatia, R.: Temporal localization of fine-grained actions in videos by domain transfer from web images. In: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, ACM (2015) 371–380

[48] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR. (2015) 2625–2634

[49] Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. arXiv preprint arXiv:1511.04119 (2015)

[50] Xu, H., Venugopalan, S., Ramanishka, V., Rohrbach, M., Saenko, K.: A multi-scale multiple instance video description network. arXiv preprint arXiv:1505.05914 (2015)

[51] Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: ICCV. (2009)

[52] Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV **61**(1) (2005) 55–79

[53] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. TPAMI **32**(9) (2010) 1627–1645

[54] Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. TPAMI (11) (1998) 1254–1259

[55] Alexe, B., Heess, N., Teh, Y.W., Ferrari, V.: Searching for objects driven by context. In: NIPS. (2012) 881–889

[56] Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML. (2015)

[57] Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. CoRR **abs/1511.04119** (2015)

[58] Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: NIPS. (2014) 2204–2212

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

[60] Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. In: ICLR. (2015)

[61] Sermanet, P., Frome, A., Real, E.: Attention for fine-grained categorization. arXiv preprint arXiv:1412.7054 (2014)

[62] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014) 580–587

[63] Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR. (2015) 759–768

[64] Gkioxari, G., Girshick, R., Malik, J.: Contextual action recognition with r* cnn. In: ICCV. (2015) 1080–1088

[65] Zhang, C., Platt, J.C., Viola, P.A.: Multiple instance boosting for object detection. In: NIPS. (2005) 1417–1424

[66] Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization. In: CVPR, IEEE (2008) 1–8

[67] Li, W., Duan, L., Xu, D., Tsang, I.W.H.: Text-based image retrieval using progressive multi-instance learning. In: ICCV, IEEE (2011) 2049–2055

[68] Song, H.O., Girshick, R., Jegelka, S., Mairal, J., Harchaoui, Z., Darrell, T.: On learning to localize objects with minimal supervision. In: ICML. (2014)

[69] Pathak, D., Shelhamer, E., Long, J., Darrell, T.: Fully convolutional multi-class multiple instance learning. arXiv preprint arXiv:1412.7144 (2014)

[70] Wu, J., Yinan, Y., Huang, C., Kai, Y.: Deep multiple instance learning for image classification and auto-annotation. In: CVPR, IEEE (2015) 3460–3469

[71] Lu, X., Lin, Z., Shen, X., Mech, R., Wang, J.Z.: Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. In: ICCV. (2015) 990–998

[72] Girshick, R.: Fast r-cnn. In: ICCV. (2015) 1440–1448

[73] Jain, M., van Gemert, J., Jegou, H., Bouthemy, P., Snoek, C.G.: Action localization with tubelets from motion. In: CVPR. (June 2014)

[74] Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Learning to track for spatio-temporal action localization. In: ICCV). (December 2015)

[75] Yu, G., Yuan, J.: Fast action proposals for human action detection and search. In: CVPR. (2015) 1302–1311

[76] Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Learning to detect motion boundaries. In: CVPR. (2015) 2578–2586

[77] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015) 3431–3440