Multi-region Two-Stream R-CNN for Action Detection

Xiaojiang Peng(✉) and Cordelia Schmid

Thoth team, Laboratoire Jean Kuntzmann, Inria, Grenoble, France
{xiaojiang.peng,cordelia.schmid}@inria.fr

Abstract. We propose a multi-region two-stream R-CNN model for action detection in realistic videos. We start from frame-level action detection based on faster R-CNN, and make three contributions: (1) we show that a motion region proposal network generates high-quality proposals, which are complementary to those of an appearance region proposal network; (2) we show that stacking optical flow over several frames significantly improves frame-level action detection; and (3) we embed a multi-region scheme in the faster R-CNN model, which adds complementary information on body parts. We then link frame-level detections with the Viterbi algorithm, and temporally localize an action with the maximum subarray method. Experimental results on the UCF-Sports, J-HMDB and UCF101 action detection datasets show that our approach outperforms the state of the art with a significant margin in both frame-mAP and video-mAP.

Keywords: Action detection · Faster R-CNN · Multi-region CNNs · Two stream R-CNN

1 Introduction

Action recognition in videos has many realistic applications such as surveillance, human computer interaction, and content-based retrieval. Most research efforts have concentrated on action classification [1–4], where a class label is assigned to an entire video. However, given a video stream, actions occur at precise spatio-temporal extents. Action detection aims at determining these location, which has attracted increasing attention recently [5–8]. It is a challenging problem due to large intra-class variations, background clutter and in particular the large spatio-temporal search space.

Several previous works address only temporal localization [9–11], i.e. they only provide the start and end time of an action. State-of-the-art results are obtained with a temporal sliding window and dense trajectory features [10]. For spatio-temporal detection, several recent works extend 2D object detection models to 3D ones. For example, Tian *et al.* [5] extend the 2D deformable part model [12] to a 3D deformable part model and Wang *et al.* [6] extend poselets [13] to a dynamic poselet model. More recent works first detect actions at a frame

B. Leibe et al. (Eds.): ECCV 2016, Part IV, LNCS 9908, pp. 744–759, 2016.
DOI: 10.1007/978-3-319-46493-0_45

level by using Convolutional Neural Networks (CNNs) features and then either link them or track some selected detections to obtain video action detections [7, 8]. Thanks to the excellent performance of CNNs for object detection, these frame-level based approaches achieve state-of-the-art performance. This suggests that the quality of the frame-level action detections impacts directly the quality of action detection in videos.

Thus, a crucial point is how to improve the frame-level action detection. Weinzaepfel *et al.* [8] improve frame-level action detection by using a better proposal algorithm, i.e. EdgeBoxes [14]. Gkioxari *et al.* [15] boost R-CNN based action detection in still images by adding contextual features. Indeed, these two approaches indicate two important issues for frame-level detection: (1) high-quality proposals help CNNs to extract action representations precisely; and (2) the action representation is vital for detection.

In this paper we focus on the frame-level based action detection method, and aim to advance the state-of-the-art with respect to these two key aspects: frame-level action proposal and action representation.

Frame-level action proposal. One of the bottleneck for object detection based on region proposals is the accurate localization of these proposals [16–18]. To address this issue for action detection, we first evaluate three proposal methods for frame-level action localization on RGB data: selective search (SS) [19], EdgeBoxes (EB) [14], and region proposal network (RPN) [17]. We show that the RPN approach on appearance information achieves consistently better results than the others with higher inter-section-over-union (IoU) score. Furthermore, we extend the appearance RPN to motion RPN trained on optical flow data. We observe that motion RPN obtains high quality proposals, which are shown to be complementary to appearance RPN.

Action representation. Action representation is crucial for good performance, see for example [20, 21]. Here, we propose an improved action representation inspired by the two-stream CNNs for action classification [22] and multi-region CNNs [18]. First, we stack multiple frame optical flows for the faster R-CNN model which significantly improves the motion R-CNN. Second, we select multiple body regions (i.e., upper body, lower body and border region) for both appearance and motion R-CNN, which boosts the performance of frame-based action detection.

In summary, this paper introduces a multi-region two-stream R-CNN model for action detection with state-of-the-art results on UCF-Sports, J-HMDB and UCF101 datasets. Our contributions are as follows: (1) we introduce a motion RPN which generates high-quality proposals and is complementary to the appearance RPN. (2) We show that stacking optical flows significantly improves frame-level detections. (3) We embed a multi-region scheme in the faster R-CNN model which is shown to improve the results.

The remained of this paper is organized as follows. In Sect. 2, we review related work on action recognition and region CNNs. We introduce the two-stream R-CNN with the motion RPN and stacked optical flows in Sect. 3. Our multi-region embedded R-CNN is described in Sect. 4 and the temporal linking and localization in Sect. 5. We present experimental results in Sect. 6.

2 Related Work

Action recognition and Convolutional Neural Networks (CNNs) have been extensively studied in recent years [23,24]. This section only covers the approaches directly related to our method.

Action classification and detection. For action classification, most methods focus on how to represent the entire video [1–4]. Popular video representations are bag-of-visual-words (BoW) [25] and its variants which aggregate local video features [1], CNN representations [22,26], and slow feature representations [27]. Wang *et al.* [1] use Fisher Vectors [28] and dense trajectories with motion compensation. Peng *et al.* [3] combine this approach with stacked Fisher Vectors. Simonyan *et al.* [22] design the two-stream CNNs based on RGB data and optical flow. Karpathy *et al.* [26] explore several approaches for fusing information over time based on appearance CNN. Wang *et al.* [4] extract two-stream CNNs along dense trajectories.

For action detection, [9,10,29,30] use local features to represent actions and rely on a sliding window scheme for either temporal or spatio-temporal localization. Rodriguez *et al.* [31] and Derpanis *et al.* [32] conduct global template matching. Tran *et al.* [33] use a BoW representation and implement the optimal spatio-temporal path for action detection. Tian *et al.* [5] extend the 2D deformable part model [12] to 3D space-time volumes for action localization. Wang *et al.* [6] apply dynamic poselets and a sequential skeleton model to jointly detect actions and poses.

Region CNN for detection. Region CNN (R-CNN) [16] has achieved a significant improvement for object detection in static image. This approach first extracts region proposals using selective search [19] and rescales them to a fixed size, and then uses a standard CNN network [34] to train and extract features. The features are subsequently fed into a SVM classifier with hard negative mining and a bounding box regressor. SPP-net improved it by removing the limitation of a fixed input size with a spatial pyramid pooling strategy [35]. Fast R-CNN speeds up the R-CNN by introducing a RoI pooling scheme and training classifier and bounding box regressor simultaneously [36]. Faster R-CNN further accelerates the fast R-CNN by replacing the selective search proposal method with a region proposal network [17]. Spyros *et al.* [18] added multi-region and segmentation-aware CNN features to make the R-CNN representation more discriminative. Inspired by R-CNN, Gkioxari and Malik [7] extract proposals by using the selective search method on RGB frames and then applied the original R-CNN on per frame RGB and optical flow data for frame-level action detection, and finally link detections by the Viterbi algorithm [37] to generate action tubes. Weinzaepfel *et al.* [8] replaced the selective search method by EdgeBoxes [14] for proposal extraction, and performed tracking on selected frame-level detections.

Our work differs from the above mentioned approaches in four ways: (1) we generate rich proposals from both RGB and optical flow data by using region proposal networks; (2) we use stacked optical flows to enhance the discriminative

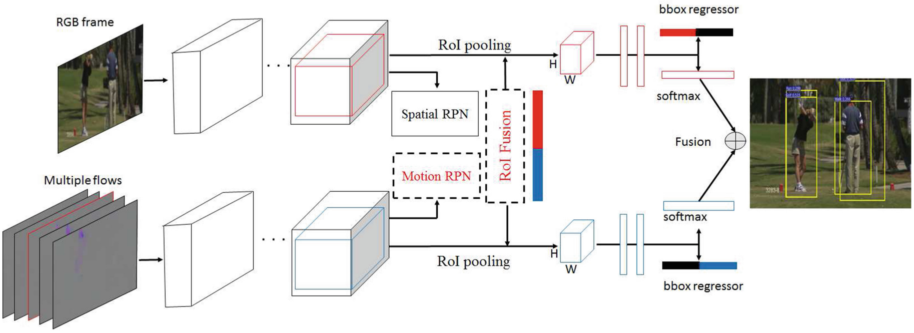

Fig. 1. Overview of our two-stream faster R-CNN. (Color figure online)

capacity of motion R-CNN; (3) we further improve the performance by embedding a multi-region scheme in faster R-CNN; and (4) we build an end-to-end multi-region two-stream CNN model for frame-level action detection.

3 End-to-end Two-Stream Faster R-CNN

Figure 1 gives an overview of our two-stream faster R-CNN (TS R-CNN) approach. Stacking optical flow has shown to be effective for CNN based action classification [22]. We believe this can also be the case for R-CNN based action detection. Our TS R-CNN takes as input an RGB frame f_t and *several optical flow maps* extracted for frame f_t and its neighboring frames (we take half of the frames before time t and half of them after). The network then processes them with several convolutional and max pooling layers, independently in the appearance and the motion stream. For each stream, the last convolutional layer is fed into an appearance or motion region proposal network and a region of interest (RoI) pooling layer. Here we introduce a RoI fusion layer, which merges the proposals from both the appearance RPN and the motion RPN. Both the appearance and the motion RoI pooling layer take all the RoIs and perform max-pooling for each of them with a $H \times W$ grid. For each stream, these fixed-length feature vectors are then fed into a sequence of fully connected layers that finally branch into a softmax layer and a bounding box regressor. The final detection results from both streams can be combined by several methods which will be evaluated in Sect. 6.3. Best performance is obtained by simply combining the softmax scores.

Training and testing. We train each of the two-stream faster R-CNNs separately. For both streams, we fine-tune the VGG-16 model [38] pre-trained on the ImageNet dataset [39]. One frame optical flow data is transformed to a 3 channel image by stacking the x-component, the y-component and the magnitude of the flow as in [8]. In case of multiple optical flow maps, where the input channel number is different from that of VGG-16 net, we just duplicate the VGG-16 filters of the first layer multiple times. We use the ground-truth bounding boxes of

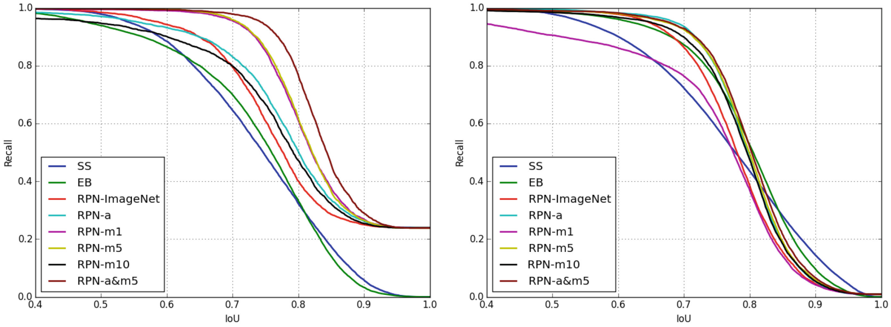

Fig. 2. Comparision of frame-level proposals on UCF-Sports and J-HMDB split 1: selective search proposals (SS), EdgeBoxes proposals (EB), RPN-ImageNet, RPN-a, RPN-m and the fusion of RPN-a and RPN-m proposals. RPN-mk indicates k frame optical flows. Left: UCF-Sports. Right: J-HMDB split 1.

the middle frame for training. For testing, we first combine the learned appearance and motion R-CNN models into one model by adding a RoI fusion layer, see Fig. 1. We then put frame-flow pairs to the end-to-end model and average the softmax scores from both streams as the final action region detection scores. The bounding box regressor is applied to corresponding RoIs of each stream (see the red and blue solid bars in Fig. 1). The concatenation of these boxes is the final detection result.

Evaluation of our action proposals. To show the quality of our motion RPN (RPN-m), we compare it to several other proposal methods. Figure 2 compares the recall over intersection-over-union (IoU) for different proposal methods described in the following. Selective search (SS) [19] generates regions by using a bottom-up grouping scheme with features from color, texture and box sizes. We keep the default setting and obtain 2 k proposals. EdgeBoxes (EB) [14] are obtained based on the observation that the number of contours entirely contained in a bounding box is indicative of the objectness. Again we use the default setting and obtain 256 proposals. The RPN method first generates several anchor boxes for each pixel with multiple scales and ratios, and then scores and regresses them with the learned features. For training RPN, positive objectness labels are obtained for those anchors that have high IoU overlap with ground-truth boxes. For the comparison, we keep RPN 300 proposals and use one scale with a fixed minimum side of 600 pixels. We also extend the RPN method to optical flow and report results for single flow and stacked flows.

Figure 2 shows that RPN-a method consistently outperforms SS and EB, i.e. it obtains best results when using RGB frames. Interestingly, on UCF-Sports it obtains perfect detections (IoU = 1) 25 % of the time (i.e., recall = 0.25 for IoU = 1). For fair comparison with SS and EB (both are non-tuned methods for action datasets), we show the results of RPN pre-trained on ImageNet as RPN-ImageNet in Fig. 2. It also consistently outperforms SS and EB on both datasets. Moreover, the motion RPN with a single frame optical flow also provides very

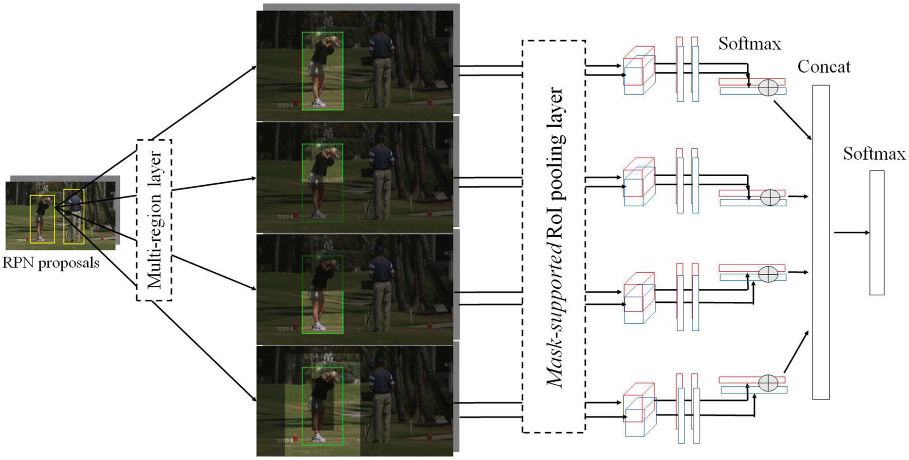

Fig. 3. Overview of the multi-region two-stream faster R-CNN architecture.

good action proposals. They are better than RPN-a on UCF-Sports, but worse on J-HMDB. This can be explained by more significant motion occurring on UCF-Sports compared to J-HMDB, which contains a number of daily activities without significant motion, such as “brush_hair”, “pour” and “wave”. The recall increases with 5 stacked flows (RPN-m5) and decreases with 10 stacked flows (RPN-m10). A possible explanation is that stacking optical flows makes the representation more discriminative, but that there is a saturation for a higher number of frames due to the non-aligned temporal boxes. Combining the proposals from both appearance and motion RPN achieves the best performance and outperforms SS and EB by a significant margin.

4 Multi-region Two-Stream Faster R-CNN

The multi-region two-stream faster R-CNN (MR-TS R-CNN) architecture is illustrated in Fig. 3. It is built on the two-stream faster R-CNN by embedding a multi-region generation layer between the RPNs and the RoI pooling layer. Given proposals from both appearance RPN and motion RPN, the multi-region layer generates 4 RoIs for each RPN proposal. We describe the 4 types of regions relevant for action representation in the following.

Original regions are the original RPN proposals. A network along this channel is guided to capture the whole action region. The network is exactly the same as the TS R-CNN. The bounding box regressor is only applied on this channel.

“*Upper half*” and “*bottom half*” regions are the upper and bottom halfs of the RPN proposals, see second and third rows in the multi-region layer in Fig. 3. Instead of left/right/upper/bottom half regions used for objects in [18], we only use the upper/bottom half regions due to the mostly symmetric vertical structure of bodies in action videos. Networks based on these parts are not only robust w.r.t occlusions but also more discriminative for action categories for which body part features are dominant. For example, "golf" and "swing_baseball" are easier to recognize by only the upper half region, while "climb_stairs" and "kick_ball" by only the bottom half region.

"*Border*" regions are rectangular rings around the original proposals. Given a RPN proposal, we generate the inner box of a border region by scaling the proposal by a factor of 0.8 and the outer box by a factor of 1.5. For the appearance stream, a network along this channel is expected to jointly capture the appearance border of human and nearby objects which may be helpful for action recognition. For motion stream, this channel has high probability to focus on the motion boundary region which was demonstrated to be very useful for handcrafted features [1].

Training. The two-stream network for original regions is copied from the one presented in the previous section. For training the two-stream networks of the other regions, we fine-tune the network of the original regions separately for each region. In particular, we only tune the fully connected layers, and fix all the *convolutional* layers as well as the RPN to ensure that all the region networks share the same proposals. Regarding the "Border" region two-stream network, we introduce a mask-supported RoI pooling layer which sets the activations inside the inner box to zero similar to [18,40]. After training the region networks, we combine them by further training another softmax layer based on the softmax layers of multi-region two-stream networks, see Fig. 3. Note that the multi-region R-CNNs share all the *conv* layers and hence the computation cost during testing increases only by a factor of 1.8.

5 Linking and Temporal Localization

Based on the above described method, we obtain frame-level action detections. In order to achieve video-level detection, we apply linking similar to [7] and temporal localization based on the maximum subarray algorithm [41].

Given two regions R_t and R_{t+1} from consecutive frames t and $t+1$, we define the linking score for an action class c by

$$s_c(R_t, R_{t+1}) = \{s_c(R_t) + s_c(R_{t+1}) + \beta\ ov(R_t, R_{t+1})\} \bullet \psi(ov), \tag{1}$$

where $s_c(R_i)$ is the class score of region R_i, ov is the intersection-over-union overlap of the two regions and β is a scalar. $\psi(ov)$ is a threshold function defined by $\psi(ov) = 1$ if ov is larger than τ, $\psi(ov) = 0$ otherwise. We experimentally observe that our linking score is better than the one in [7] and more robust due to the additional overlap constraint. After computing all the linking scores of an action, we obtain video-level action detections by determining the optimal path iteratively with the Viterbi algorithm. We finally score a video-level action detection $\mathfrak{R} = [R_1, R_2, ..., R_T]$ by $\overline{s_c(\mathfrak{R})} = \frac{1}{T}\sum_{i=1}^{T} s_c(R_i)$.

In order to determine the temporal extent of an action detection within a video track, one can apply a sliding window approach with multiple temporal scales and strides as [8]. Here we rely on an efficient maximum subarray method. Given a video-level detection $\mathfrak{R}$, we aim to find a detection from frame s to frame e which satisfies the following objective,

$$s_c(\mathfrak{R}^{\star}_{(s,e)}) = \underset{(s,e)}{\operatorname{argmax}}\{\frac{1}{L_{(s,e)}}\sum_{i=s}^{e} s_c(R_i) - \lambda\frac{|L_{(s,e)} - L_c|}{L_c}\}, \quad (2)$$

where $L_{(s,e)}$ is the track length and L_c is the average duration of class c on the training set. We propose to approximately solve this objective by three steps: (1) subtract from all the frame-level action scores the video-length action score $s_c(\mathfrak{R})$, (2) find the maximum subarray of the subtracted array by using Kadane's algorithm [41], (3) extend or shorten the optimal range to L_c. Our solution searches the track only once. For each video-length action detection, we only keep the best extent as spatio-temporal detection. Note that the threes-step heuristic is an approximation to Eq. (2), and step (3) sets the length of the optimal tube from step (2) to the average length to avoid degenerate solutions.

6 Experiments

In this section, we first present the details of datasets and the evaluation metrics and describe the implementation details. We then evaluate our method comprehensively and compare to the state of the art.

6.1 Datasets and Evaluation Metrics

In our experiments, we evaluate action detection on three datasets: UCF-Sports, J-HMDB and UCF-101. We briefly review them in the following and present the metrics used for evaluation.

UCF-Sports [31] contains 150 short videos of 10 different sport classes. Videos are truncated to the action and bounding boxes annotations are provided for all frames. We use the standard training and test split defined in [31].

J-HMDB [20] consists of 928 videos for 21 different actions such as brush hair, swing baseball or jump. Video clips are restricted to the duration of the action. Each clip contains between 15 and 40 frames. Human silhouettes are annotated for all frames. The ground-truth bounding boxes are inferred from the silhouettes. There are 3 train/test splits and evaluation averages the results over the three splits.

UCF-101 [42] is dedicated to action classification with more than 13000 videos and 101 classes. For a subset of 24 labels and 3207 videos, the spatio-temporal extents of the actions are annotated. All experiments are performed on the first split only. In contrast to UCF-Sports and J-HMDB where the videos are truncated to the action, UCF-101 videos are longer and the localization is both spatial and temporal.

Evaluation metrics. We use three metrics in our experiments: (i) *frame-AP*, the average precision of detection at the frame level as in [7]; (ii) *video-AP*, the

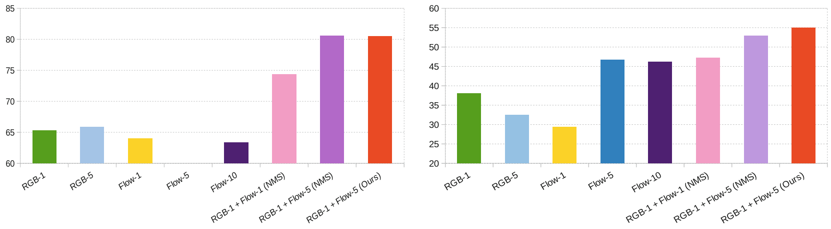

Fig. 4. Evaluation of different frames types (RGB and flow), number of frames (x = 1, 5, 10) used for detection and combination strategies (NMS or score combination–ours). Left: UCF-Sports. Right: J-HMDB split 1.

average precision at the video level as in [7,8]. We fix the IoU threshold to [0.2, 0.5] for frame-AP and video-AP measurement on the UCF-Sports and J-HMDB, and [0.05, 0.1, 0.2, 0.3] on UCF101.

6.2 Implementation Details

We implement our method based on the Caffe open source toolbox[1]. Optical flow is estimated using the online code from Brox *et al.* [43]. For both appearance and motion R-CNN, we use the same setting except that motion R-CNN uses $128, 128, 128$ as the mean data values. Similar to [36], we use a single sample, i.e. either a single image or a stacked optical flow map annotated with ground-truth boxes at every training iteration. When fine-tuning the VGG-16 model, we only update layers from *conv*3_1 and up as observed to be efficient in [36]. For action region proposal network training, we set the regions with IoU larger than 0.7 as positive regions, and the regions with IoU less than 0.3 as negative regions. For the classification part of faster R-CNN, we use 256 proposals with a quarter of them as positive bounding boxes from the RPN, where the IoU of a positive box is larger than 0.5 and of a negative between 0.1 and 0.5. When training the two-stream R-CNN on UCF-Sports and J-HMDB, we initialize the learning rate to 10^{-3}, decrease it to 10^{-4} after 50 K iterations, and stop training after 70 K iterations. When training the *multi-region* two-stream R-CNN, we only fine-tune the fully-connected layers of the TS R-CNN model and set the learning rate to 10^{-4}, change it to 10^{-5} after 7 K iterations, and stop after 10 k iterations. We double the mentioned iterations on the UCF101 dataset empirically since it is a much larger dataset. The grid of RoI pooling layer is fixed to 7×7. The dropout rates of fully connected layers are set to 0.5 in all cases. The threshold τ of function $\psi(ov)$ is fixed to 0.2 empirically.

6.3 Evaluation of Multi-region Two-Stream Faster R-CNN

In this section, we first evaluate our method for frame-level detection with respect to four aspects: RGB/flow stacking, stream combination, multi-scale training and

[1] https://github.com/rbgirshick/py-faster-rcnn.

Table 1. Evaluation of different training and testing scales. All detections from different scales are combined by the NMS, and RGB-1 and Flow-5 streams are combined by score averaging. We report results for UCF-Sports and J-HMDB, split 1.

		RGB-1		Flow-5		RGB-1 + Flow-5	
Test scales	*Train* scales	UCF-Sports	J-HMDB	UCF-Sports	J-HMDB	UCF-Sports	J-HMDB
{600}	{600}	65.30	38.05	74.24	46.71	-	-
	{480, 600, 800}	68.07	38.71	73.62	47.74	-	-
{480, 600, 800}	{600}	68.47	39.90	**76.77**	47.05	-	-
	{480, 600, 800}	**69.29**	**40.02**	75.81	**48.60**	**82.30**	**56.60**

testing and multi-region scheme. We then present the spatio-temporal detection at the video level and the action classification results based on detection.

RGB and optical flow faster R-CNN with several frames. We compare appearance and motion faster R-CNN with one or multiple frames for frame-level detection (mean AP), see Fig. 4. For this evaluation, the training/testing scale is fixed to 600 which corresponds to the shorter side of an *input image*. We can observe that appearance R-CNN extracted for one frame (RGB-1) outperforms motion R-CNN extracted for one frame (Flow-1) on both UCF-Sport and J-HMDB. Increasing the number of frames for the appearance model (RGB-5) does not improve the performance. However, using 5 frames for flow significantly improves the performance for motion R-CNN, i.e. we gain 10.27 % on UCF-Sports and 17.39 % on J-HMDB split 1. This is mainly due to the fact that motion information from one frame is not discriminative enough, see a similar observation for action classification in [22]. Stacking more flows (Flow-10) decreases the result significantly on UCF-Sports, and slightly on J-HMDB partly due to the degraded proposals as mentioned in Sect. 3. We observe that the decrease in performance is more important on the strongly moving actions such as “Diving”, “Swinging at the high bar”, “Kicking”, and “Jump”. This can be explained by the fact the stacking does not align the actors and hence the detected bounding boxes are more imprecise. In summary, Flow-5 performs best and is complementary to RGB-1. We discuss different combination schemes next.

Two streams combination. We explore two schemes for combining the appearance and motion R-CNN: box-level non maximum suppression (NMS) and score fusion with a RoI fusion layer (our end-to-end pipeline, see Fig. 1). The NMS method perform detection for appearance and motion R-CNN separately, and then fuses all the detected bounding boxes from both streams with NMS. As shown in Fig. 4, for the fusion of RGB-1 and Flow-5 streams, the score fusion (indicated by “Ours”) obtains 2.02 % improvement over the NMS fusion on J-HMDB and performs on par on UCF-Sports. Compared to the NMS fusion, the score fusion uses both appearance and motion information for bounding box scoring which is more discriminative. In the remained of this paper, we use score fusion for the combination of appearance and motion R-CNNs, and use “RGB-1 + Flow-5” for two-stream R-CNN by default.

Multi-scale training and testing. An action can occur on arbitrary scales. Here we explore robustness to scale changes by using multi-scale training and

Table 2. Per-class frame-AP of individual regions and multi-region two-stream faster R-CNN on UCF-Sports.

Region	Diving	Golf	Kicking	Lifting	Riding	Run	SkateBoarding	Swing1	Swing2	Walk	*mAP*
Org	94.68	66.34	72.06	98.53	97.54	**84.04**	**59.67**	79	98.20	72.87	82.30
Upper half	95.84	80.44	33.61	99.10	97.51	81.45	17.08	60.78	98.08	69.87	73.38
Bottom half	88.34	48.12	67.62	96.86	**97.61**	79.75	57.98	**84.16**	98.12	72.91	79.15
Border	95.91	69.54	66.74	**99.95**	97.02	80.18	50.53	52.14	98.15	**76.52**	78.67
Multi-region	**96.12**	**80.47**	**73.78**	99.17	97.56	82.37	57.43	83.64	**98.54**	75.99	**84.51**
Gkioxari *et al.* [7]	75.8	69.3	54.6	99.1	89.6	54.9	29.8	88.7	74.5	44.7	68.1
Weinzaepfel *et al.* [8]	60.71	77.55	65.26	100.00	99.53	52.60	47.14	88.88	62.86	64.44	71.9

Table 3. Per-class frame-AP of individual regions and multi-region two-stream faster R-CNN on J-HMDB (average on 3 splits).

Region	brushHair	catch	clap	climbStairs	golf	jump	kickBall	pick	pour	pullup	push	run	shootBall	shootBow	shootGun	sit	stand	swingBaseball	throw	walk	wave	*mAP*
Org	70.5	**39**	60.1	60.2	99.3	11.2	**35.9**	**59.1**	**97**	97.4	78	32.4	**52.9**	90.1	52.4	29.2	**49.3**	53.9	**27.8**	**60.5**	**38.1**	56.9
Upper half	**77.9**	33.1	60.8	48.6	94.7	5.40	20	44	89.6	93.9	62.9	24.5	46.4	83.5	57.9	**36.7**	38.2	**62.6**	20.5	49.4	29.3	51.4
Bottom half	36.4	29.4	32.3	**68.1**	97.3	6.90	32.9	26.5	87	93.7	67.1	31.9	33	72.6	21.7	20.8	41.3	44.3	18	38.1	24.9	44.0
Border	64.9	34.6	58.9	52.6	99.5	11.4	35	49.6	94.6	95.2	71.3	32.4	46.5	83.8	50.9	25.1	46.9	45.6	22.9	56.1	32.3	52.9
Multi-region	75.8	38.4	**62.2**	62.4	**99.6**	**12.7**	35.1	57.8	96.8	97.3	**79.6**	**38.1**	52.8	**90.8**	**62.7**	33.6	48.9	62.2	25.6	59.7	37.1	**58.5**
[7]	65.2	18.3	38.1	39.0	79.4	7.3	9.4	25.2	80.2	82.8	33.6	11.6	5.6	66.8	27.0	32.1	34.2	33.6	15.5	34.0	21.9	36.2
[8]	73.3	34.0	40.8	56.8	93.9	5.9	13.8	38.5	88.1	89.4	60.5	21.1	23.9	85.6	37.8	34.9	49.2	36.7	16.8	40.5	20.5	45.8

testing. We fix the scale to 600 for single scale training/testing, and to {480, 600, 800} for the multi-scale case. The results are shown in Table 1. The results of multi-scale training is on par of single-scale training when testing on one scale only. However, multi-scale training with multi-scale testing achieves consistent better results than the other settings. In particular, it improves the single-scale training and testing by 4 % for the RGB-1 R-CNN model on UCF-Sports. Our two-stream R-CNN with multi-scale setting obtains 82.3 % and 56.6 % on UCF-Sports and J-HMDB, respectively. In the remained of the paper, we fix the setting to multi-scale training and testing.

Multi-region R-CNN. For the multi-region evaluation, we use the two-stream model RGB-1 + Flow-5 and the multi-scale setting for all part R-CNN models. We report the per-class results of our region R-CNN models in Table 2 for UCF-Sports and in Table 3 for J-HMDB. Among all the R-CNN models on both datasets, the *Org* R-CNN achieves the best performance in mean AP, which indicates the whole body is essential for an action. On UCF-Sports, the *Bottom half* and *Border* models get similar results as the *Org* model, while the *Upper half* model is worse than the *Org* model by a margin of 9 %. In contrast, on J-HMDB the *Bottom half* model gets the worst result and the other region models obtain similar results with the *Org* model. This reflects the different type of actions in the two datasets, i.e., J-HMDB is dominated by upper body actions (everyday actions), while for UCF-Sports the bottom part of the action is most characteristic (sport actions). The multi-region two-stream R-CNN (MR-TS R-CNN) model improves the *Org* R-CNN model by 2.21 % and 1.6 % on UCF-Sports and J-HMDB datasets, respectively. It also outperforms the state-of-the-art methods [7,8] with a large margins on both datasets.

Furthermore, we observe that individual part R-CNN models perform better than the *Org* model for some actions. For example, the *Upper half* model gains 14.1 % for "Golf" on UCF-Sports and 8.7 % for "swingBaseball" on J-HMDB over the *Org* model. Also, the *Bottom half* model gains 5.16 % for "Swing 1" on

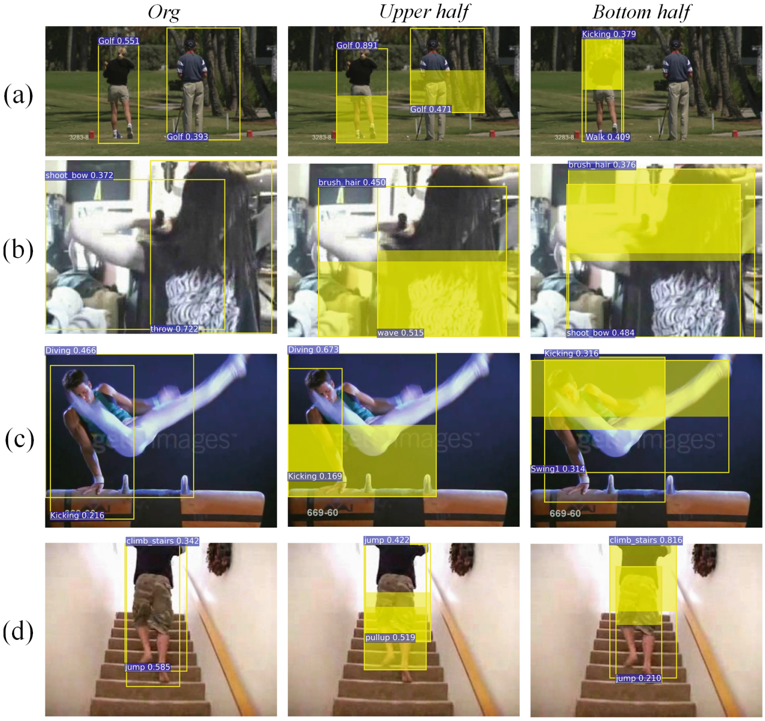

Fig. 5. Examples of action detection with different region R-CNN models. We only show the top two detections after performing NMS. Overlayed yellow regions indicate the regions invisible for the part R-CNN models. (Color figure online)

UCF-Sports and 7.9 % for "climbStairs" on J-HMDB. Figure 5 illustrates this with a few examples. Actions in row (a) and (b) are better detected by *Upper half* R-CNN model, while row (c) and (d) by *Bottom half* R-CNN model. We can observe that the detected boxes and their scores vary significantly between the different models. By focusing on the bottom part, the example of "climb_stairs" (row d, column 3) gets a high confidence detection due to the discriminative cue of stairs and legs.

Linking and temporal localization. We evaluate our linking and temporal localization methods for both the two stream R-CNN model and its multi-region version. Table 4 shows the video mAP results with IoU threshold of δ on UCF-Sports, J-HMDB, and UCF101 datasets. Both of our approaches obtain excellent video-level performance on these datasets which is mainly due to the high-quality frame-level detections. We obtain 94.82 % and 70.88 % on UCF-Sports and J-HMDB (split 1) with our linking method, respectively. The corresponding numbers are 94.81 % and 68.97 with the linking method in [7]. Results improve on J-HMDB, but are the similar for UCF-Sports, where detections are near perfect. Multi-region TS R-CNN consistently outperforms the original TS R-CNN model on J-HMDB and UCF101, and performs similarly on UCF-Sports. The lack in

Table 4. Video mAP on UCF-Sports, J-HMDB and UCF101 (split 1) with variant IoU thresholds.

	UCF-Sports		J-HMDB		UCF101 (with temporal loc)				UCF101 (w/o)
δ	0.2	0.5	0.2	0.5	0.05	0.1	0.2	0.3	0.2
TS R-CNN	94.82	**94.82**	71.1	70.6	54.13	49.51	41.17	31.13	40.67
MR-TS R-CNN	**94.83**	94.67	**74.3**	**73.09**	**54.46**	**50.39**	**42.27**	**32.70**	40.95

Table 5. Classification results on UCF-Sports and J-HMDB by detection.

	FAT [7]	IDT+FV [1]	P-CNN (w/o GT) [21]	TS R-CNN	MR-TS R-CNN
UCF-Sports	-	88.0	-	91.49	**95.74**
J-HMDB	62.5	65.9	61.1	70.52	**71.08**

improvement on UCF-Sports might be explained by the mistakes in the spatio-temporal annotation, which explains why the classification performance actually does improve for UCF-Sports. Note that we only perform temporal localization on UCF101. Nevertheless, most of the action classes cover almost the entire video. By temporal localization with MR-TS R-CNN model, we observe a gain of 1.3 % in video-mAP for IoU threshold of 0.2, but on actions “Basketball”, “BasketballDunk”, and “CricketBowling”, we gain 19.6 %, 16.2 %, and 9.6 % respectively.

Classification by detection. Similar to [7], our approach can be also extended to action classification of the videos. We leverage the best action track (i.e., the track with maximum action score) in a video to predict the action label. Table 5 reports the average class accuracy on UCF-Sports and J-HMDB. Both of our models achieve outstanding performance, with the multi-region version improving the results in both cases. In particular, our MR-TS R-CNN model obtains 95.74 % and 71.08 % on UCF-Sports and J-HMDB, respectively. The results are significantly better than those of the IDT method [1] and the pose-based CNN method [21] which perform only classification. This suggests that classification can be improved by *precise localization* and *detection-aware* features.

6.4 Comparison to the State of the Art

We conclude the experimental evaluation with a comparison to the state of the art in Table 6. In both frame-level and video-level mAP, our TS R-CNN already outperforms the state-of-the-art results on both UCF-Sports and J-HMDB, and is on par with [8] on UCF101. In particular, our MR-TS R-CNN approach outperforms the state of the art by 12.6 %, 12.7 % and 4.79 % in frame-mAP, 4.3 %, 12.4 % and 0.2 % on UCF-Sports, J-HMDB and UCF101, respectively. Both [7,8] also make use of frame-level action detection with R-CNN. Weinzaephel *et al.* [8] select the top two frame-level detections for each class from the entire video and then track with them based on class-level and instance-level scores.

Table 6. Comparison to the state of the art on three datasets. The IoU threshold δ for frame-mAP is fixed to 0.5.

		UCF-Sports		J-HMDB		UCF101 (split 1)			
	δ	0.2	0.5	0.2	0.5	0.05	0.1	0.2	0.3
Video-mAP	Gkioxari *et al.* [7]	-	75.8	-	53.3	-	-	-	-
	Weinzaepfel *et al.* [8]	-	90.5	63.1	60.7	54.3	51.7	46.8	37.8
	Yu *et al.* [44]	-	-	-	-	49.9	42.8	26.5	14.6
	Our TS R-CNN	94.8	**94.8**	71.1	70.6	54.1	49.5	41.2	31.1
	Our MR-TS R-CNN	**94.8**	94.7	**74.3**	**73.1**	**54.5**	50.4	42.3	32.7
Frame-mAP	Gkioxari *et al.* [7]	68.1		36.2		-			
	Weinzaepfel *et al.* [8]	71.9		45.8		35.84			
	Our TS R-CNN	82.3		56.9		**39.94**			
	Our MR-TS R-CNN	**84.5**		**58.5**		39.63			

This allows them to increase the video-mAP relative to their frame mAP, in particular for difficult datasets such as UCF101. Yet, such an additional tracking step is complementary to our approach. Compared to [7,8], our method benefits from two key points: (1) the high-quality proposals from both appearance and motion RPN and (2) the discriminative frame-level action representation based on stacked optical flows and multiple parts.

7 Conclusion

This paper introduces a multi-region two-stream R-CNN action detection approach, which takes full advantage of three recent methods, namely faster R-CNN, two-stream CNNs with optical flow stacking and multi-region CNNs. We propose a novel framework for action detection which builds on these methods. It significantly outperforms the state of the art [7,8]. In our experiments on UCF101, we observed that a limitation lies in handling low-quality videos and small bounding boxes, which will be addressed in future work.

Acknowledgments. This work was supported in part by the ERC advanced grant ALLEGRO, the MSR-Inria joint project, a Google research award, a Facebook gift, the Natural Science Foundation of China (No. 61502152) and the Open Projects Program of National Laboratory of Pattern Recognition. We gratefully acknowledge the support of NVIDIA with the donation of GPUs used for this research.

References

1. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV, pp. 3551–3558 (2013)
2. Jain, A., Gupta, A., Rodriguez, M., Davis, L.: Representing videos using mid-level discriminative patches. In: CVPR, pp. 2571–2578 (2013)

3. Peng, X., Zou, C., Qiao, Y., Peng, Q.: Action recognition with stacked fisher vectors. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 581–595. Springer, Heidelberg (2014)
4. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: CVPR, pp. 4305–4314 (2015)
5. Tian, Y., Sukthankar, R., Shah, M.: Spatiotemporal deformable part models for action detection. In: CVPR, pp. 2642–2649 (2013)
6. Wang, L., Qiao, Y., Tang, X.: Video action detection with relational dynamic-poselets. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 565–580. Springer, Heidelberg (2014)
7. Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR, pp. 759–768 (2015)
8. Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Learning to track for spatio-temporal action localization. In: ICCV, pp. 3164–3172 (2015)
9. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization of actions with actoms. PAMI **35**(11), 2782–2795 (2013)
10. Oneata, D., Verbeek, J., Schmid, C.: Efficient action localization with approximately normalized Fisher vectors. In: CVPR, pp. 2545–2552 (2014)
11. Escalera, S., et al.: ChaLearn looking at people challenge 2014: dataset and results. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8925, pp. 459–473. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16178-5_32
12. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR, pp. 1–8 (2008)
13. Bourdev, L., Malik, J.: Poselets: body part detectors trained using 3D human pose annotations. In: ICCV, pp. 1365–1372 (2009)
14. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 391–405. Springer, Heidelberg (2014)
15. Gkioxari, G., Girshick, R., Malik, J.: Contextual action recognition with R*CNN. In: ICCV, pp. 1080–1088 (2015)
16. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587 (2014)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS, pp. 91–99 (2015)
18. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware CNN model. In: ICCV, pp. 1134–1142 (2015)
19. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**(2), 154–171 (2013)
20. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.: Towards understanding action recognition. In: ICCV, pp. 3192–3199 (2013)
21. Chéron, G., Laptev, I., Schmid, C.: P-CNN: pose-based CNN features for action recognition. In: ICCV, pp. 3218–3226 (2015)
22. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS, pp. 568–576 (2014)
23. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. (CSUR) **43**(3), 16 (2011)
24. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
25. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)

26. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR, pp. 1725–1732 (2014)
27. Sun, L., Jia, K., Chan, T.H., Fang, Y., Wang, G., Yan, S.: DL-SFA: deeply-learned slow feature analysis for action recognition. In: CVPR, pp. 2625–2632 (2014)
28. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
29. Laptev, I., Pérez, P.: Retrieving actions in movies. In: ICCV 2007, pp. 1–8 (2007)
30. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR, pp. 2442–2449 (2009)
31. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR, pp. 1–8 (2008)
32. Derpanis, K.G., Sizintsev, M., Cannons, K., Wildes, R.P.: Efficient action spotting based on a spacetime oriented structure representation. In: CVPR, pp. 1990–1997 (2010)
33. Tran, D., Yuan, J., Forsyth, D.: Video event detection: from subvolume localization to spatiotemporal path search. PAMI **36**(2), 404–416 (2014)
34. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
35. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. PAMI **37**(9), 1904–1916 (2015)
36. Girshick, R.: Fast R-CNN. In: ICCV, pp. 1440–1448 (2015)
37. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. Inf. Theory **13**(2), 260–269 (1967)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 (2014)
39. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
40. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: CVPR, pp. 3992–4000 (2015)
41. Bentley, J.: Programming pearls: algorithm design techniques. Commun. ACM **27**(9), 865–873 (1984)
42. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402 (2012)
43. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
44. Yu, G., Yuan, J.: Fast action proposals for human action detection and search. In: CVPR, pp. 1302–1311 (2015)