Pattern Recognition 79 (2018) 32–43

Contents lists available at ScienceDirect

Pattern Recognition

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

Multi-stream CNN: Learning representations based on human-related regions for action recognition

Zhigang Tu[a], Wei Xie[b], Qianqing Qin[c,*], Ronald Poppe[d], Remco C. Veltkamp[d], Baoxin Li[e], Junsong Yuan[f]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore
[b] School of Computer, Central China Normal University, LuoyuRoad 152, Wuhan, China
[c] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, LuoyuRoad 129, Wuhan, China
[d] Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, Utrecht, The Netherlands
[e] School of Computing, Informatics, Decision System Engineering, Arizona State University, AZ 85287, USA
[f] Computer Science and Engineering department, State University of New York at Buffalo, US

ARTICLE INFO

Article history:
Received 13 May 2017
Revised 10 January 2018
Accepted 24 January 2018

Keywords:
Convolutional Neural Network
Action recognition
Multi-Stream
Motion salient region

ABSTRACT

The most successful video-based human action recognition methods rely on feature representations extracted using Convolutional Neural Networks (CNNs). Inspired by the two-stream network (TS-Net), we propose a multi-stream Convolutional Neural Network (CNN) architecture to recognize human actions. We additionally consider human-related regions that contain the most informative features. First, by improving foreground detection, the region of interest corresponding to the appearance and the motion of an actor can be detected robustly under realistic circumstances. Based on the entire detected human body, we construct one appearance and one motion stream. In addition, we select a secondary region that contains the major moving part of an actor based on motion saliency. By combining the traditional streams with the novel human-related streams, we introduce a human-related multi-stream CNN (HR-MSCNN) architecture that encodes appearance, motion, and the captured tubes of the human-related regions. Comparative evaluation on the JHMDB, HMDB51, UCF Sports and UCF101 datasets demonstrates that the streams contain features that complement each other. The proposed multi-stream architecture achieves state-of-the-art results on these four datasets.

1. Introduction

The amount of video data available is experiencing explosive growth due to the ubiquity of digital recording devices and the popularity of video sharing web sites. The recognition of the actions and activities of the people in these videos is one of the long-standing research topics in the computer vision community. It has been extensively investigated in recent years [1–4]. Action recognition is the process of labeling video frames with action labels, and it enables a computer to automatically recognize high-level human behaviors. Video-based action recognition has attracted increasing interest in recent years due to its wide range of applications, such as human computer interaction, video gaming interfaces like Microsoft Kinect [5], video surveillance, and health care.

Video-based action recognition is a challenging problem [1]. First, we face intra-class (different performances within an action class) and inter-class (similarities between different action classes) variations. Second, environments and recording settings can vary significantly. For example, human localization becomes much harder in a dynamic, cluttered environment. Third, humans typically move considerably, and isolating a specific action performance in time is often not trivial. Actions to be recognized are often assumed to be already segmented in time, which requires a separate segmentation process. This may not always be feasible due to the difficulty in appearance/motion segmentation. In this work, we are concerned with how to accurately detect the human and the primary moving body part under challenging conditions. This is a fruitful way toward robust feature extraction, which eventually determines the result of action recognition.

Recently, following their success in static image classification [6], convolutional neural networks (CNNs) have been extended to take into account motion information for use in action recognition [7–10]. Human actions in videos can naturally be viewed as 3D spatio-temporal signals, characterized by the temporal evolution of

[*] Corresponding authors.
E-mail addresses: TUZG@ntu.edu.sg (Z. Tu), XW@mail.ccnu.edu.cn (W. Xie), Qqqin@lmars.whu.edu.cn (Q. Qin), R.W.Poppe@uu.nl (R. Poppe), R.C.Veltkamp@uu.nl (R.C. Veltkamp), Baoxin.Li@asu.edu (B. Li), jsyuan@buffalo.edu (J. Yuan).

https://doi.org/10.1016/j.patcog.2018.01.020
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

visual appearance governed by motion [11]. Approaches have been proposed to learn spatio-temporal features to represent spatially and temporally coupled action patterns. One representative work is [12], which presents a two-stream CNN architecture. One stream models the appearance representation, while the other stream models the motion representations from optical flow. These two representations are complementary, and better performance is obtained when combining them.

Features that encode the region of interest (ROI) of a human as a whole are considered global features. The ROI is normally extracted by leveraging background subtraction or tracking [1]. As a result, global representations are sensitive to variations in viewpoint, background motion, noise and illumination changes. Recently, [13] applied selective search [14] to select regions in each frame, and discard those regions void of motion according to a motion saliency measure (MSM) based on optical flow. While this approach overcomes the dependency on background subtraction and tracking, it has some drawbacks. First, there is no suitable method to select the saliency threshold, which directly affects the selection of regions that are salient in shape and motion. Second, actions with subtle motions could be missed. Third, the selected regions are not necessarily spatially coherent.

Cheron et al. [7] take a different approach to extract person-centered features by first estimating the location of the body joints. The regions corresponding to four body parts (right/left hand, upper/full body) are used to extract motion and appearance features. Each body part, and the full ROI are encoded as streams in a multi-stream CNN. The main challenge is to estimate body poses [15]. It is suggested that the current robustness of algorithms is insufficient for action recognition [16]. Another limitation is the extensive computation load.

The work presented in this paper builds on these previous works, but we aim at robustness and accuracy by improving the selection of human-related regions (HRs) and the processing of the deep features of the multiple streams. Specifically, we extend the end-to-end convolutional two-stream network (TS-Net) [17] by adding four streams that focus on the entire human-body (termed R1, see Fig. 1) and the primary moving body part (R2). Our multi-stream network thus contains three TS-Nets: three motion-related and three appearance-related streams. R1 is extracted based on an improved Block-sparse Robust Principal Component Analysis (IB-RPCA) method and R2 is detected via a motion saliency measure that depends on the obtained object information of IB-RPCA. To effectively combine the six streams, we adopt the spatio-temporal 3D convolutional fusion approach of [17]. We term our approach the human-related multi-stream CNN (HR-MSCNN).

The Block-sparse Robust Principal Component Analysis (B-RPCA) technique [18] is employed to detect humans. It is able to cope with background motion, illumination changes, noise and poor image quality in a unified framework. Motion saliency estimation is applied to refine the foreground regions, which enforces spatial coherence. To improve the performance of B-RPCA, we add a velocity angle measure to improve the consistency of the motion direction. The detected ROI in each frame is the entire human. According to the obtained motion information, we propose a motion saliency measure to extract one part where the movement is most distinctive. This secondary ROI can convey highly discriminative information, complementary to the ROI of the whole human.

This paper extends [9], and presents the following novel contributions:

- We present an end-to-end convolutional neural network that considers the regions of a detected person as well as discriminative regions of motion related to the performance of an action.
- We propose a novel method to obtain static and dynamic features and exploit a new fusion method to efficiently combine the information from the six streams for final action prediction.
- We test different object detection and optical flow methods, and reveal that both the quality of the detected human region and the input optical flow significantly affect the recognition performance.
- We extensively evaluate various aspects of our approach and compare it with previous algorithms on common benchmark datasets (i.e., UCF101 [19] and HMDB51 [20]), where we achieve state-of-the-art recognition performance.

The code of our approach is publicly available.[1]

The remainder of the paper is organized as follows. We review related work on action recognition in Section 2. Section 3 describes the HR-MSCNN framework. The detection of HRs is discussed in Section 4. We evaluate our method in Section 5 and conclude in Section 6.

2. Related work

Local, hand-crafted features have long been the dominant technique for action recognition [11,21]. Local features do not require the detection of the body and are robust to illumination changes, video noise, and cluttered backgrounds. Since video can inherently be considered as a 3D spatio-temporal signal, extending 2D spatial image descriptors to 3D spatio-temporal video descriptors has been extensively investigated. Among these hand-crafted spatio-temporal features, improved dense trajectories (iDT) [21] has achieved outstanding performance. It combines three low-level descriptors: histograms of oriented gradients (HOG), histograms of optical flow (HOF), and motion boundary histograms (MBH). Once extracted, the local features are then encoded by bag-of-words or Fisher vectors to produce a global video representation. One problem with these local features is that they lack semantics and discriminative capacity [8]. For more realistic and complex human actions, the performance of these descriptors often degrades sharply due to the challenges of intra-class and inter-class variations [22].

Learned feature representations are a promising alternative to hand-crafted features. Since the seminal work of [6], learning visual features with CNN has been intensively studied for many recognition tasks including image classification [23], scene recognition [24], object detection [25] and face recognition [26]. Attempts have also been made to learn representations with CNNs for tasks involving the temporal domain, including action recognition [7,13,27,28]. Deep-learned features have shown advantages over hand-crafted features. The reason is that large-scale training datasets allow deep architectures to learn a hierarchy of semantically related convolution filters that increase the discriminative capability.

There are two primary strategies used in deep learning to extract features from video frames. First, 3D CNN [29] learns convolution kernels in the spatial and temporal domains by extending conventional 2D CNN architectures [6] to 3D. To process the 3D spatio-temporal signals more effectively, Sun et al. [22] exploited a new deep architecture. Compared to training the 2D conventional kernels, it is more complex to train the 3D kernels, and the current size and variation of training datasets for action recognition in videos is insufficient to guarantee good performance.

The second strategy is a two-stream CNN [12], which achieved state-of-the-art performance [8,17,30]. Gkioxari and Malik [13] leveraged object proposals to localize actions. Object proposals are captured per-frame using the selective search strategy [14].

[1] https://github.com/ZhigangTU/HR-MSCNN.

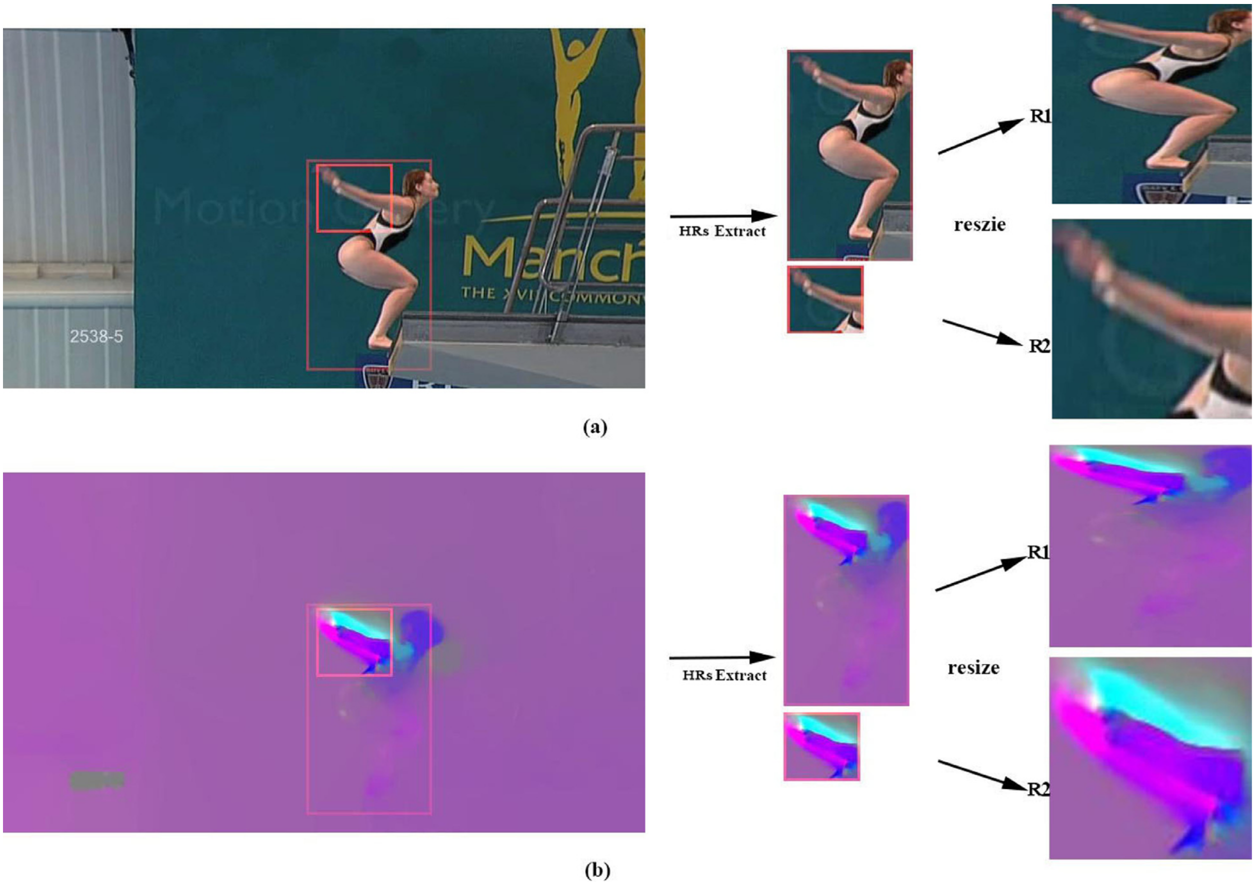

Fig. 1. HR extraction and resizing on (a) the RGB input image and (b) the motion field. Region 1 (R1) contains the full body, region 2 (R2) the motion salient body part.

These are scored by applying spatio-temporal features extracted from the two-stream CNN, and linked over time through the video.

Motion regions associated to moving body parts can transmit highly discriminative information, which will result in a more discriminative action representation [28]. In the context of action recognition, HRs obtained using a pose estimator have been used by Cheron et al. [7]. Gkioxari et al. [31] exploited a body part-based method by leveraging CNN features. Gkioxari et al. [32] applied Region-based Convolutional Networks (R-CNN) [25] to utilize multiple regions for classification. Singh et al. [33] exploited a Multi-Stream Bi-directional Recurrent Neural Network (MSB-RNN) for fine-grained action detection in videos. Based on the detected bounding box of the person, two person-centered streams are obtained. The method faces two drawbacks. First, the human-centered region alone is not sufficiently informative to extract features for action recognition. Second, the tracker performs poorly in capturing human bodies in videos.

In this paper, we focus on robust detection of HRs because of their demonstrated merits in containing discriminative information. Spatio-temporal features are extracted at these detected regions, similar to the two-stream CNN. We further adopt the spatio-temporal architecture for TS-Net by Feichtenhofer et al. [17] to better extract features for action recognition by exploiting a 3D convolutional fusion followed by 3D pooling at the final convolutional layer.

To handle the inability of the TS-Net to model long-range temporal structure, Long Short-Term Memory (LSTM) can be integrated in the CNN. Ng et al. [34] presented two methods to enable the TS-Net to make use of full length videos for video classification. The first method is temporal feature pooling that exploits max-pooling of local information over time. The second method is a RNN that applies LSTM with hidden states evolving with every subsequent frame. Donahue et al. [35] designed a Long-term RNN (LRCNN) architecture, where LSTM's long-term temporal recursion models are connected to CNN models. Wang et al. [30] designed a temporal segment network (TSN) to extract long-term temporal structure for action recognition. Additionally, they demonstrated that exploiting more input modalities to construct more streams leads to higher recognition rates. These models that allow for action recognition in long videos can be integrated with the approach presented in this paper.

3. Multi-stream CNN: Features extraction from HRs

We propose a multi-stream CNN that consists of three two-stream networks (TS-Nets), to extract features in those image regions that contain discriminative information of the action. Fig. 2 outlines the framework of our HR-MSCNN architecture schematically. We detect two complementary HRs in terms of the IB-RPCA technique and motion saliency separately (see Fig. 1). In each frame, the primary HR (R1), a bounding box around the detected human, is extracted using IB-RPCA. The secondary HR (R2), a region located within the primary HR, is captured using our motion saliency measure. The third region (R3) contains the entire input RGB image or optical flow image, which supplies the global spatial context. Based on these three regions, three motion streams are formulated from the optical flow field, and three appearance streams are obtained from the RGB image. Descriptors are extracted from the streams and fused for video-based action recognition. Specially, spatio-temporal 3D convolution fusion is adopted from the convolutional TS-Net of [17].

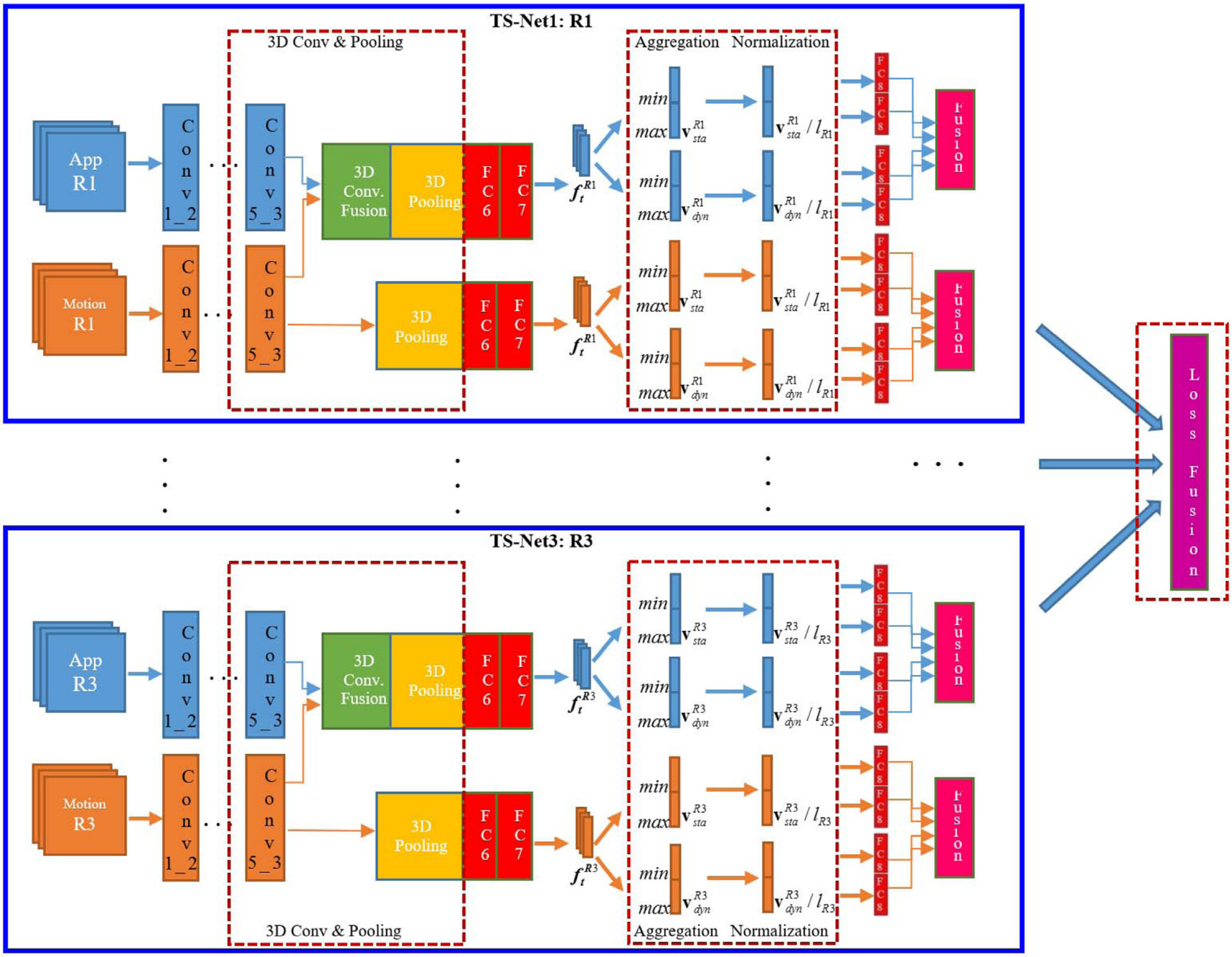

Fig. 2. Schematic representation of the HR-MSCNN framework for the three regions R1, R2 and R3. From top to bottom are three TS-Nets, one for each region. Using R1 as our example, the three parts of the network, from left to right: 1) **3D convolution fusion and pooling**: at the last convolution layer, we fuse the two streams into a spatio-temporal stream by utilizing 3D convolution fusion followed by 3D pooling; we perform 3D pooling in the motion network separately. 2) **Static and dynamic aggregation**: after the second fully-connected layer, static frame descriptors f_t^{R1} are aggregated across all frames into static video descriptor v_{sta}^{R1}, and temporal differences of f_t^{R1} are aggregated into dynamic video descriptor v_{dyn}^{R1}. 3) **Loss fusion**: the predictions of R1, R2, and R3 are averaged for final action prediction.

3.1. CNN descriptors

To capture the features within our HR-MSCNN, we use the MatConvNet toolbox [36] for the implementation of convolutional networks. A brief description of our training process follows.

3.1.1. Step 1: Processing input data

To construct a motion-CNN, the optical flow [37] is first calculated for each pair of successive frames using the method of [38]. The *x*- and *y*-components, and the magnitude of the flow are rescaled to the range of [0, 255]: $[\widehat{u}, \widehat{v}] = \gamma[u, v] + 128$, where $\gamma = 16$ is the rescale factor. Values outside the range are set to 0. Then, same as [13], these components are stacked to form a 3D image as the input for the motion-CNN. During training, for each selected HR, we resize it to 224 × 224 to fit the CNN input layer. To construct a spatial-CNN, each selected HR in the RGB image is also resized to 224 × 224.

3.1.2. Step 2: Selecting and training CNN models

One motion model and one appearance model, pre-trained by [17] based on the VGG-16 model [39] are employed to learn six stream representations of the RGB and optical flow images. The VGG-16 model with 13 convolutional and 3 fully-connected layers performs better than shallow models such as VGG-f [40] with 5 convolutional and 3 fully-connected layers. This results in more accurate results (see Table 6).

3.1.3. Step 3: Spatial and temporal fusion

To take advantage of both motion and appearance information, we adopt the convolutional TS-Net of [17]. By employing a novel convolutional fusion layer (3D convolution and 3D pooling) [17] between the motion-network and appearance-network, we are able to learn correspondences between abstract spatial and temporal features. In other words, our network can recognize what moves where.

3.1.4. Step 4: Aggregation

As shown in Fig. 2, our approach employs a novel static and dynamic aggregation approach to aggregate deep features. We first form a video descriptor by aggregating all frame descriptors f_t^{Rj} where *Rj* represents one HR and *t* is the frame. One frame descriptor f_t^{Rj} is a 4096-dimensional vector, i.e., the output of the second fully-connected layer (FC7). We then formulate the min and max aggregation by calculating the minimum and maximum values for each descriptor dimension i ($i \in \{1, \ldots, n\}, n = 4096$) over all T frames of the input video:

$$\begin{aligned} m_i &= \min_{1 \le t \le T} f_t^{Rj}(i) \\ M_i &= \max_{1 \le t \le T} f_t^{Rj}(i) \end{aligned} \tag{1}$$

For the two static video descriptors v_{sta}^r, we concatenate the time-aggregated frame descriptors:

$$\begin{aligned} v_{sta}^{Rj}(min) &= [m_1, \ldots, m_n]^T \\ v_{sta}^{Rj}(max) &= [M_1, \ldots, M_n]^T \end{aligned} \tag{2}$$

The dynamic video descriptors v_{dyn}^{Rj} are obtained by concatenating the minimum $\triangle m_i$ and maximum $\triangle m_i$ aggregations of $\triangle f_t^{Rj}$

$$\begin{aligned} v_{dyn}^{Rj}(min) &= [\triangle m_1, \ldots, \triangle m_n]^T \\ v_{dyn}^{Rj}(max) &= [\triangle M_1, \ldots, \triangle M_n]^T \end{aligned} \tag{3}$$

where $\triangle f_t^{Rj} = f_{t+\triangle t}^{Rj} - f_t^{Rj}$, $\triangle t = 4$ is the time interval.

Finally, for the appearance stream of region *Rj*, the scores of the two static video descriptors $v_{sta}^{Rj}(min)$ and $v_{sta}^{Rj}(max)$ are weighted with a ratio $v_{sta}^{Rj}(min) : v_{sta}^{Rj}(max) = 1 : 4$. The scores of two dynamic video descriptors are fused in the same way. The scores of v_{sta}^{Rj} and v_{dyn}^{Rj} are combined according to $v_{sta}^{Rj} : v_{dyn}^{Rj} = 1 : 1$. For the motion stream of region *Rj*, the prediction is computer similarly. Then, different from [17], we fuse the predictions of the appearance stream and the motion stream with a ratio 1 : 1.5 as in [30], because features learned from the motion stream are typically more informative than those of the appearance stream.

3.1.5. Step 5: Fusion of predictions from HRs

To obtain the final action classification, we average the predictions of the three different HRs, R1, R2 and R3.

4. Detection of human-related regions

Detecting moving objects is an extensively investigated subject [41] and significant progress has been achieved. Most existing techniques still face some challenges with complex natural data. In this work, the B-RPCA technique [18] is employed. We propose a modified version of B-RPCA, which we call improved B-RPCA (IB-RPCA), to detect the foreground human in the input image. In addition, motion saliency is exploited to extract one motion salient region (MSR) corresponding to the human body detected from the previous step.

4.1. Detection of humans

Since actions are strongly associated with actors, features computed from the detected region around the actor can help in the recognition of actions, because the location variation from the input representation will be removed [33]. This is especially useful for actions that are related to human body motion only [42] such as “Walk-Front” in the UCF Sports dataset, “Sit” and “Stand” in JHMDB or to human-object interaction [42] such as “Riding-Horse” in the UCF Sports dataset or “Swing-baseball” in JHMDB.

4.1.1. B-RPCA

To deal with the challenges in detecting foreground moving objects, Gao et al. [18] imposed constraints on the background. The background can be identified according to a low-rank conditional matrix. Mathematically, the observed video frames can be considered as a matrix **M**, which is a sum of two matrices: a low-rank matrix **L** that denotes the background, and a sparse outlier matrix **S** that consists of the moving objects. The foreground moving objects can be captured by solving the decomposition using robust principal component analysis (RPCA) [43]. More recently, [18] introduced a feedback scheme and proposed a block-sparse RPCA (B-RPCA) technique that consists of a hierarchical two-pass process to handle the decomposition problem. B-RPCA consists of three major steps, summarized below to facilitate the later discussion of our improvements to B-RPCA.

Step 1: First-pass RPCA. In this step, a first-pass RPCA at a sub-sampled resolution is applied to quickly detect the likely foreground regions:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \| \boldsymbol{L} \|_* + \lambda \| \boldsymbol{S} \|_1, \qquad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} \tag{4}$$

where $\|L\|_*$ denotes the nuclear norm of the background matrix **L** and λ is a regularization parameter that controls the number of outliers in the RPCA decomposition. The appropriate value is $\lambda = 1/\sqrt{\max(m, n)}$ (where $m \times n$ denotes the dimensions of matrix **M**). Eq. (4) presents a convex optimization problem, which can be solved by applying the augmented Lagrange multiplier (ALM) [44]. Through this first-pass RPCA, outliers can be identified and stored in outlier matrix **S**.

Step 2: Motion saliency estimation (MSE). A motion consistency strategy is used to assess the motion saliency of the detected foreground regions and the probability that a block contains moving objects. Pixels within the blocks captured in the first round of RPCA are tracked by optical flow. After tracking, dense point trajectories are extracted. First, trajectories shorter than 10 frames are removed. Second, we estimate the motion saliency of the remaining trajectories according to the consistency of the motion direction [45]. This is beneficial for two reasons: (1) foreground objects that move in a slow but consistent manner can be better identified; (2) small, local motion that comes from inconsistent motions of the background can be discarded. Most of the non-stationary background motions that were identified and stored in the outlier matrix **S** in the first step are removed or suppressed.

Step 3: Second-pass RPCA. In this step, the λ value is reset according to the motion saliency [18]. This ensures that changes derived from the foreground motion can be completely transferred to the outlier matrix **S**. The second pass RPCA is implemented as:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \| \boldsymbol{L} \|_* + \sum_i \lambda_i \| \mathbf{P}_i(\mathbf{S}) \|_F, \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} \tag{5}$$

where $\| \cdot \|_F$ denotes the Frobenius norm of the matrix. $\mathbf{P}_i$ is an operator that unstacks columns of **S** and returns a matrix that represents block *i*. The ALM algorithm is again employed to solve Eq. (5).

4.1.2. Improved B-RPCA (IB-RPCA)

The MSE operation is effective in filtering out or suppressing the non-stationary background motions in the video as long as the object keeps moving in a way that is spatio-temporally constant. However, if the object occasionally stops or moves with a velocity close to zero, all foreground object pixels will be removed in the second step of B-RPCA. MSE also requires a significant amount of time to compute. In particular, when the background contains non-stationary motions, the computation time will increase exponentially. To overcome these difficulties with B-RPCA, we propose the following improvements:

1) We relax the minimal length of each trajectory from 10 to 5 frames. This way, sudden stops in the motion of the actor will be significantly reduced. To avoid noise arising from the background, we add the motion derivative constraint similar to MBH [21]. By calculating derivatives of the optical flow components u and v, the background motion due to locally constant camera motion will be excluded.

2) We enhance the consistency measure of the motion direction. Not only the direction of u and v along the trajectory, but also the variation in the direction should be considered. Hence, we add a velocity angle measure:

$$\triangle\theta = \arctan(u_{t+1}/v_{t+1}) - \arctan(u_t/v_t) \in [-\pi/4, \pi/4] \tag{6}$$

where $[u_t, v_t] \neq 0$. Just like the motion direction consistency operation, this velocity angle measure is also conducted at positions where the velocity is non-zero.

3) If u_t and v_t satisfy either Eqs. (7) or (8), we consider the actor to be stationary between frames t and $t+1$. We then only perform the first step of B-RCPA to detect the actor in the RGB images.

$$(range(u_t) < T_d) \wedge (range(mGflow) < T_m) \tag{7}$$

$$(range(v_t) < T_d) \wedge (range(mGflow) < T_m) \tag{8}$$

Here, $range(u_t)$ denotes the difference between the maximum and minimum values of u_t. $mGflow = \sqrt{(u_t)_x^2 + (u_t)_y^2 + (v_t)_x^2 + (v_t)_x^2}$ is the magnitude of the spatial gradient of the optical flow (u_t, v_t). The thresholds T_d and T_m are empirically set to 0.5. For one motion field, if each of its motion vectors has a motion displacement smaller than a half pixel, we consider this motion field as static.

Algorithm 1 summarize our IB-RPCA method, and Algorithm 2 describe the most important modification: the improved Motion Saliency Estimation. In particular, the details of Algorithm 2 are expressed as follows: $X_{l,j}, X_{l,j+1}, \ldots, X_{l,k}$ represents the consecutive position of points along the l-th trajectory from frame j to frame k. Short trajectories with $k - j \leq 5$ are removed. The motion saliency of the remaining trajectories are calculated according to the consistency of the motion direction in step (2) of Algorithm 2. P_u and P_v denote the number of frames in the trajectory that have motion in the positive u and v directions, respectively. N_u and N_v represent the corresponding counts in the negative directions.

Algorithm 1 Improved B-RPCA (IB-RPCA).

```
Input: Video frames
Output: Object boundaries of each frame
  1) Step 1: First-pass RPCA
  2) If $u_t$ or $v_t$ does not satisfy Eq. 7 or Eq. 8 continue
    else
      Step 2: improved Motion Saliency Estimation (Algorithm 2)
      Step 3: Second-pass RPCA
    endif
```

4.2. Selecting motion salient regions (MSR) of humans

As suggested in [7,13,27], selecting suitable MSRs of the actor body is essential because the movement of some body parts is potentially helpful in improving action recognition. This is because actions are characterized by the temporal evolution of appearance governed by actor motion [11]. Some actions are characterized by the discriminative motion of body parts, such as, “brush-hair” and “clap” in the JHMDB dataset (see Fig. 5). Therefore, it is important to identify the body parts that are salient due to actor motion. Based on the captured human information of the IB-RPCA, we introduce a motion saliency measure to select a MSR where the motion is most distinguishable, see also Fig. 3.

First, we extract MSR candidates from the detected human body according to a conditional measure defined as:

$$LabH \wedge (mGflow > AmGflow) \wedge (mflow > Amflow) \tag{9}$$

Algorithm 2 Improved motion saliency estimation.

```
Input: Trajectories $X_{l,j}, X_{l,j+1}, \ldots, X_{l,k}$
Output: Selected Trajectories with Motion Saliency
  1) Initialization: $P_u = N_u = P_v = N_v = 0$
  2) Count the consistency of horizontal flow $u$:
    for $t = j : k$;
      if $(u_t(X_{l,t}) > 0)$ & $(\Delta\theta \in [-\pi/4, \pi/4])$
        $\rightarrow P_u = P_u + 1$
      if $(u_t(X_{l,t}) < 0)$ & $(\Delta\theta \in [-\pi/4, \pi/4])$
        $\rightarrow N_u = N_u + 1$
    endfor
  3) Count the consistency of vertical flow $v$, similar to step (2) above.
  4) If either $P_u, P_v, N_u$ or $N_v$ is larger than $0.8(k - j)$, then the trajectory $l$ is labeled as belonging to anobject with salient motion. Otherwise, the trajectory is marked as inconsistent.
```

and

$$(|u| > Au) \vee (|v| > Av) \tag{10}$$

where *LabH* denotes the labels of the human body region that were detected in the previous step. *AmGflow* is the mean of *mGflow* over the spatial domain. *mflow* is the magnitude of the optical flow (u, v), and *Amflow* is the mean of *mflow. Au* and *Av* are the means of the horizontal flow u and the vertical flow v, respectively.

We then discard motion salient candidates whose spatial area is small. Different body parts have different motion patterns. In addition, some background motions around the human body may be inaccurately identified by IB-RPCA. We thus attempt to remove these incorrectly captured background motions using the following measure:

$$MSR(i) > \tau \tag{11}$$

where i is the index of MSR candidates and τ is a threshold. If the area of $MSR(i)$ is smaller than τ, it will be removed. In this paper, we empirically set $\tau = 100$ pixels (e.g., the area of a 10 × 10 region). The third subfigure in Fig. 3 shows that most of the outliers are suppressed.

Finally, we capture the two largest MSR candidates in each frame, where the MSR candidates are detected according to the MATLAB built-in function *Bwlabel* that labels connected components on our computed binary image. We employ the motion saliency measure of [13] to select the final MSR by comparing the normalized magnitude of the optical flow between these two candidates:

$$flow_m(R_i) = \frac{1}{|R_i|} \sum_{j \in R_i} flow(j) \tag{12}$$

where $flow_m(R_i)$ is the normalized magnitude of the optical flow in the i-th MSR candidate. The MSR candidate with largest $flow_m(R_i)$ is selected.

5. Experiments

In this section, we evaluate our HR-MSCNN method on four publicly available datasets: UCF Sports [46], UCF101 [19], JHMDB [47], and HMDB51 [20]. These challenging datasets are often used for benchmarking, and we compare against state-of-the-art algorithms. We assess whether IB-RPCA is effective in detecting the foreground human under complex situations, and whether we can extract useful MSRs. We further evaluate whether the secondary HR is complementary to the human body region, and whether it is able to enhance the performance. In addition, we evaluate the effect of the quality of the detected human region and the input optical flow. Testing in MATLAB, with one Titan-X GPU, our method

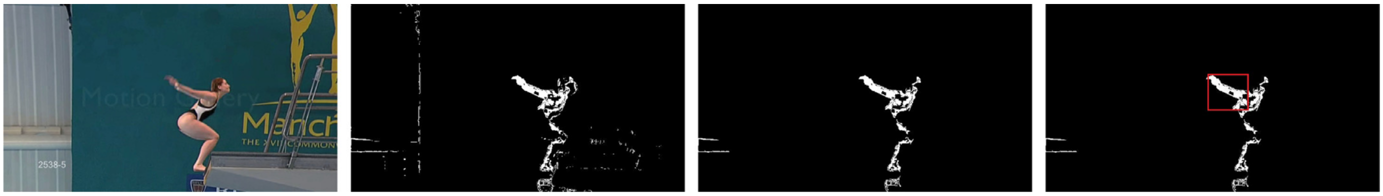

Fig. 3. Selection of the MSR in the human body. Left to right: Input RGB image, extracted MSR candidates, small MSR candidates discarded, and the most salient motion region.

Table 1
Performance with human body regions (R1) captured by different object detection methods on the JHMDB dataset. GT denotes the ground truth bounding box.

Methods	R-CNN [25]	Faster R-CNN [48]	B-RPCA [18]	IB-RPCA (Ours)	GT
Accuracy(%)	56.45	57.03	56.16	58.12	60.05

can process about 15 frames per second on average on the UCF101 dataset.

5.1. Datasets

UCF Sports contains 150 videos of sports broadcasts that are captured in cluttered, dynamic environments. There are 10 action classes and each video corresponds to one action. Note that the other three datastes share this characteristic that each video corresponds to a single action. We use the training/testing split from [46].

UCF101 contains 13,320 videos categorized into 101 action classes. They cover a large range of activities such as sports and human-object interaction. It is a challenging dataset as the captured videos vary significantly in scale, illumination, background and camera motion. The evaluation is reported as mean accuracy across three splits.

HMDB51 consists of 6766 action videos of 51 action categories. The videos are collected from a wide range of sources, including movies and online videos. We follow the suggested evaluation protocol and report the average accuracy over the three splits.

JHMDB, a subset of HMDB51, contains 928 videos of 21 different actions. Each action is present in at least 36 and at most 55 video clips. Three training/testing splits are provided in JHMDB, and the evaluation results are averaged over the three splits.

5.2. Evaluation of the effect of human detection quality

To investigate the influence of the captured human body on the accuracy of action recognition, we compare four object detection methods including widely used image-based approaches R-CNN [25] and Faster R-CNN [48], video-based technique B-RPCA [18] and our improved version IB-RPCA. We evaluate on the JHMDB dataset as it contains actor bounding boxes per frame. Table 1 shows that the action recognition performance largely depends on the quality of the captured actor region. IB-RPCA outperforms the original B-RPCA by 1.96%, and outperforms R-CNN [25] and Faster R-CNN by 1.67% and 1.09%, respectively. The results of R-CNN [25] and Faster R-CNN demonstrate that the object detection quality directly impacts the recognition accuracy. Although Faster R-CNN can extract multiple objects in a static image, it performs poorly in capturing a single actor consistently across the whole video. This is due to the lack of enforced temporal consistency. Actor-tube detector IB-RPCA is therefore more suitable to construct a human region-based stream for action recognition.

5.3. Evaluation of the effect of optical flow quality

In the two-stream framework [12], Simonyan and Zisserman demonstrated that temporal ConvNets trained on optical flow outperform spatial ConvNets significantly. This confirms that motion information plays a critical role in action recognition. To deeper investigate the importance of motion, we test the influence of the quality of the input optical flow on the UCF Sports dataset. We compare four optical flow algorithms. MPEG flow [49], which is efficiently obtained from video decompression directly without additional cost, is sparse and low resolution. The optical flow algorithm of [38] is relatively fast and has decent accuracy. It has been widely used in action recognition. The WLIF-Flow algorithm [50] is more accurate than [38] but is also computationally more expensive. Both methods [38] and [50] obtain sub-pixel accuracy. We also test EpicFlow [51], which performs better than [50] on the large-displacement MPI-Sintel dataset [52] and performs worse than [50] on the small-displacement Middlebury dataset [53]. This method is as efficient as [38], but it only delivers pixel-level accuracy.

Table 2
Mean average precision (mAP) of HR-MSCNN on the UCF Sports dataset, when using each specified optical flow method for the motion inputs.

Methods	MPEG [49]	EpicFlow [51]	[38]	WLIF-Flow [50]
mAP(%)	80.36	96.21	97.53	**98.05**

As shown in Table 2, the worst quality optical flow, MPEG flow, produces the lowest accuracy on action recognition. WLIF-Flow improves the recognition performance of MPEG flow by about 18%, and it also performs slightly better than [38]. Although the optical flow quality of EpicFlow [51] is good, its action recognition accuracy is lower than both [38] and [50]. We believe this is because EpicFlow does not provide sub-pixel motion information, and it lacks the ability to deal with small motions. We conclude that high quality optical flow is essential for high-quality video representations for action recognition. However, it does not seem to be true that more accurate optical flow leads to an action recognition performance gain. When efficiency is not an issue, a flow algorithm that is able to preserve small motion details while also handling large displacements is most suitable for action recognition. Considering both the optical flow quality and the computational time, we select the optical flow algorithm of [38] for the remaining experiments.

5.4. Evaluation on UCF sports

We now turn to the evaluation of different datasets, starting with UCF Sports. Fig. 4 shows the two detected HRs on 6 action categories. These actions are recorded in various challenging conditions, such as with multiple actors, fast motion, large displacements, occlusion, motion blur, and illumination changes. Our approach can deal with these challenges.

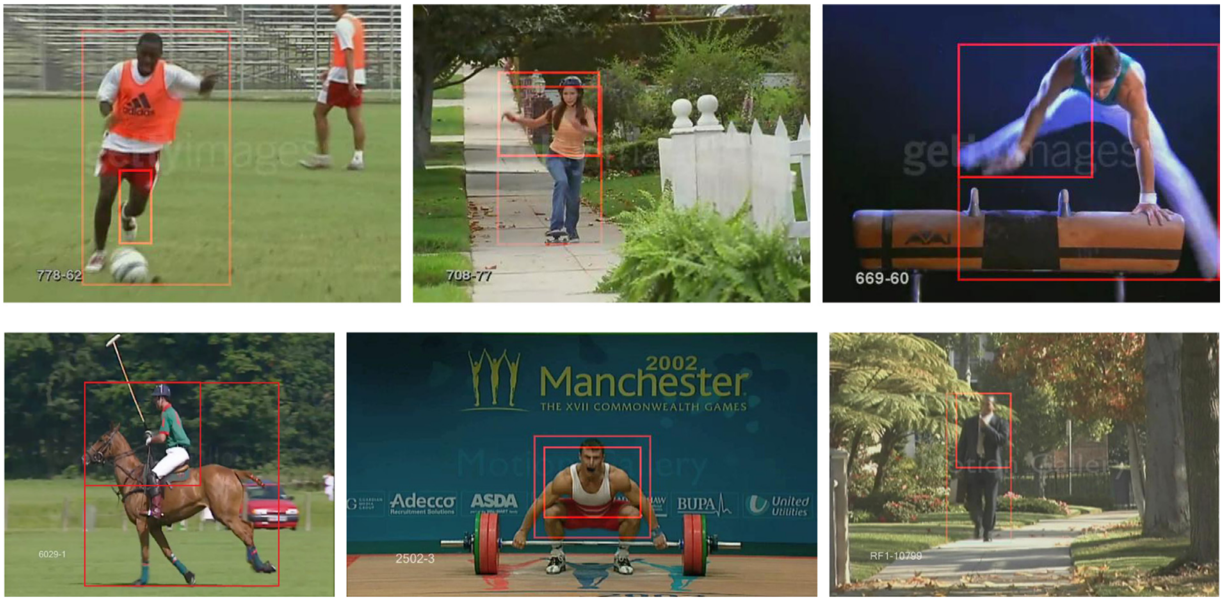

Fig. 4. Results on UCF Sports. In each image, the larger rectangle corresponds to the extracted foreground human body, and the smaller one corresponds to the secondary HR.

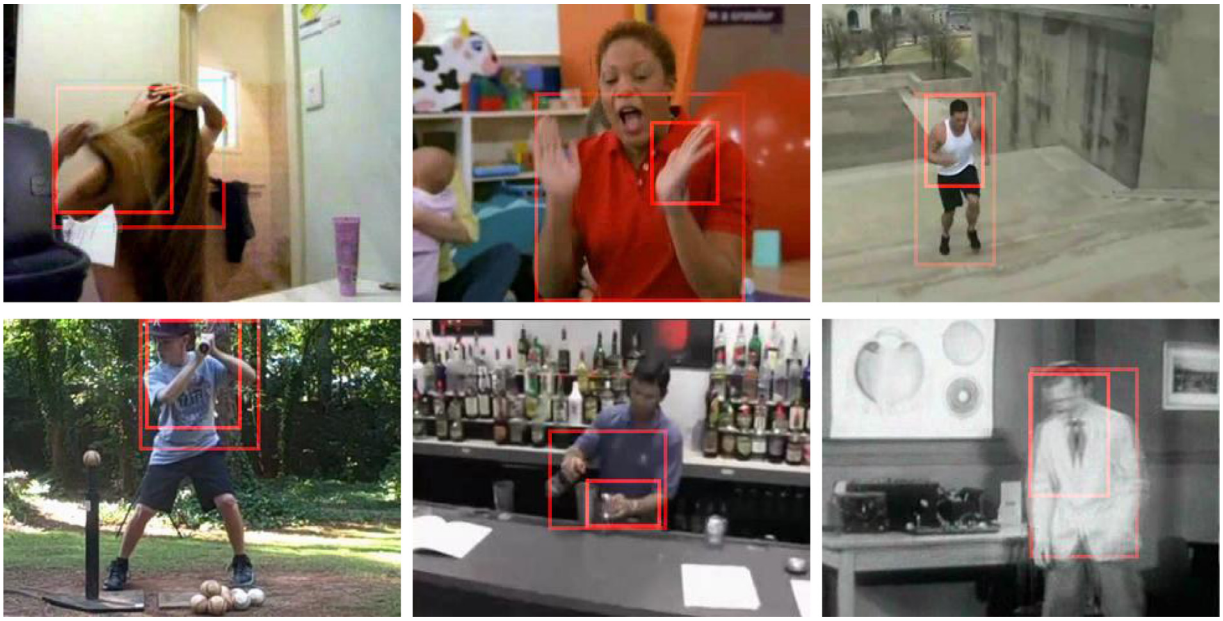

Fig. 5. Results on JHMDB. The larger rectangle corresponds to R1, and the smaller to R2.

Table 3 summarizes the results of HR-MSCNN using different combinations of regions. When comparing the first with the second row, we find that R1 and R2 contain different feature information. For example, the “SkateBoarding” action is poorly recognized using region R1 alone, but region R2 provides more information. The action can be recognized accurately when using the three regions together. The scene context of the action is preserved owing to the full frame (R3) as input. R1 and R2 emphasize the actor and its motion salient part. In this way, background noise is reduced, thus discriminative features are obtained from these two semantic cues. When integrating them all, both location-independent and location-dependent feature information can be employed. Consequently, a gain is achieved, where the video-level mean average precision (mAP) on the UCF Sports dataset is increased to 97.53%.

5.5. Evaluation on JHMDB

Fig. 5 shows the two detected HRs on examples from the JHMDB dataset. Our detectors again perform well in complex, realistic situations. Table 4 shows that different regions play different roles in action recognition, and combining them increases the action recognition performance. The results of 74.6% mAP from *All* regions outperforms other approaches by more than 4%. In terms of accuracy, our approach with *All* regions again performs best (71.17%, see Table 5).

5.6. Comparison with the state-of-the-art

In this section, we compare the proposed HR-MSCNN approach with several state-of-the-art methods. We select methods that are

Table 3
Mean average precision (mAP) of HR-MSCNN on the UCF Sports dataset using different region inputs, where R1 denotes the human body region, R2 represents the motion salient part of the actor, and R3 is the full frame.

Regions	Div.Side	Golf	Kick.	Lift.	Rid.Horse	Run	Ska.Board.	Swing1	Swing2	Walk	mAP
R1	100.0	100.0	100.0	100.0	100.0	63.89	25.00	87.67	100.0	100.0	87.65
R2	100.0	100.0	52.50	100.0	100.0	63.89	87.67	87.67	100.0	100.0	89.17
R3	100.0	100.0	52.50	100.0	100.0	63.89	87.67	63.89	100.0	100.0	86.80
R1+R2	100.0	100.0	52.50	100.0	87.67	63.89	87.67	63.89	100.0	100.0	85.56
R1+R3	100.0	100.0	100.0	100.0	100.0	63.89	52.50	63.89	100.0	100.0	88.03
R2+R3	100.0	100.0	100.0	100.0	100.0	29.17	63.89	100.0	100.0	100.0	89.31
All	100.0	100.0	100.0	100.0	100.0	87.67	100.0	87.67	100.0	100.0	**97.53**

Table 4
Mean average precision (mAP) of HR-MSCNN on the JHMDB dataset using different region inputs.

Regions	R1	R2	R3	R1 + R2	R1 + R3	R2 + R3	All
Brushhair	80.1	93.6	71.4	81.5	80.8	86.8	92.2
Catch	56.3	56.8	56.2	52.8	57.7	54.0	57.5
Clap	65.6	73.7	55.6	57.8	56.4	68.8	73.7
Climbstairs	61.8	50.1	67.6	54.2	56.8	56.1	59.2
Golf	88.8	80.8	91.3	90.9	91.3	91.3	91.3
Jump	47.5	56.4	52.7	57.3	59.8	60.1	64.6
Kickball	52.1	56.3	59.1	55.5	60.1	56.2	63.7
Pick	60.1	61.8	60.1	64.6	56.3	64.0	72.2
Pour	88.0	82.1	97.0	95.7	94.2	93.8	98.8
Pullup	98.8	98.2	100.0	100.0	100.0	100.0	100.0
Push	86.8	93.0	88.0	91.2	92.1	88.8	97.6
Run	56.3	67.7	52.8	57.3	55.5	69.4	72.4
Shootball	42.4	51.2	45.9	51.6	52.2	52.3	55.0
Shootbow	82.0	72.9	93.0	91.7	93.6	92.3	94.9
Shootgun	63.7	66.3	75.1	67.0	67.7	71.8	74.2
Sit	74.8	68.5	65.6	66.2	61.8	66.4	66.8
Stand	75.3	70.3	68.8	67.9	68.5	68.2	74.8
Swingbaseball	66.8	61.8	42.4	50.5	52.2	57.7	65.6
Throw	10.7	22.5	31.6	21.1	22.5	27.3	43.4
Walk	86.4	65.6	87.3	87.8	88.0	88.0	88.2
Wave	59.7	42.4	52.1	67.5	67.0	67.3	61.8
mAP	66.9	66.3	67.3	68.1	68.3	70.5	**74.6**

Table 5
Accuracy (%) of HR-MSCNN on the JHMDB dataset using different region inputs.

	R1	R2	R3	R1 + R3	R2 + R3	All
Accuracy(%)	62.98	62.91	63.56	65.83	68.32	**71.17**

Table 6
Performance comparison with state-of-the-art methods on the JHMDB dataset.

Methods	DT [54]	SR-CNNs [42]	P-CNN [7]	A-Tubes [13]	MSR-CNN [9]	Ours
Accuracy(%)	56.60	65.51	61.10	62.50	66.02	**71.17**

derived or extended from the classical TS-Net [12], and which were tested on the JHMDB dataset. In addition, we compare against the hand-crafted dense trajectories (DT) of Wang et al. [54]. Table 6 shows that our method performs best among them. The accuracy of our method is 14.57% better than DT [54] (71.17% *vs* 56.60%), is 5.66% more accurate than SR-CNNs [42] (65.51%), is 10.1% more precise than Pose-CNN (61.10%), is 8.67% more accurate than Action Tubes (62.50%), and outperforms MSR-CNN [9] (66.02%) by 5.15%.

From the comparison with DT, we find that the deep-learned features outperform the hand-crafted features for action recognition. We now focus on various deep-learning approaches. The two-stream semantic region based CNNs (SR-CNNs) algorithm is similar to our method. It incorporates semantic regions that are detected by Faster R-CNN [48] into the original two-stream CNNs. Since this method uses all detected regions, not only the human body but also other foreground and background regions are used. Features extracted in these other regions may negatively impact the performance of SR-CNNs. In contrast, our method focuses on the human body region and the motion salient body part, where the features in these HRs are beneficial for the task of action recognition.

For Pose-CNN, estimating human poses is a challenging task and the pose-estimator used in Pose-CNN does not always perform well. The object detectors we applied can handle all kinds of challenges in realistic scenes, and can extract the moving actor body as well as one of its primary moving body part precisely. Our method also outperforms Action Tubes [13]. The experientially selected α in Action Tubes is a fixed constant, which might not be optimal for different kinds of videos. Not only some fine-scale moving objects would be removed, but also some large-scale moving objects in challenging conditions would be incorrectly captured. Compared to MSR-CNN [9], the network proposed in this paper performs much better for three reasons. First, the new framework is end-to-end learnable. Second, a deeper CNN architecture, VGG-16, is applied to replace the shallow network VGG-f in [9]. Finally, a spatiotemporal 3D Convolutional fusion method is introduced for fusion.

We also compare our method to recent TS-Net related algorithms on UCF101 and HMDB51. We also integrated the popular hand-crafted iDT features [21] with our deep-learned HR-MSCNN features using late fusion. Table 7 shows that this approach outperforms all other methods. Without the integration of iDT, our method performs better than all of them except the TSN (3 modalities) [30]. It was already shown that the BNInception [57] architecture used in TSN is deeper than the VGG-16 network we employ. Compared to the convolutional TS-Net [17], we improve by 1.2% on UCF101 and 1.5% on HMDB51. This demonstrates that the exploited HRs provide additional information. The performance gain of 5.7%

Table 7
Performance comparison with state-of-the-art methods on the UCF101 and HMDB51 datasets.

Methods	UCF101	HMDB51
iDT+FV [21]	85.9	57.2
TDD+FV [8]	90.3	63.2
LRCNs [35] (CaffeNet)	82.9	–
RCNN-LSTM [34] (GoogLeNet)	88.6	–
Multilayer Multimodal Fusion [55] (VGG16, C3D)	91.6	61.8
Dynamic Image Networks [56] (CaffeNet)	89.1	65.2
TS-Net [12] (VGG-M)	88.0	59.4
SR-CNNs [42] (VGG-16)	92.6	–
Conv. TS Fusion [17] (VGG-16)	92.5	65.4
TSN(3 modalities) [30] (3 modalities, BNInception)	94.2	69.4
Ours (VGG-16)	93.7	66.9
Ours (VGG-16 + iDT)	**94.5**	**69.8**

on UCF101 and 7.5% on HMDB51 over the original TS-Net [12] is significant. We expect this is caused by the more effective feature selection of the spatio-temporal 3D convolutional fusion technique utilized in our method, as well as the deeper CNN architecture we have used. On UCF101, our result is 1.1% more accurate than SR-CNNs [42], which reveals that the regions detected by our proposed IB-RPCA are more relevant than those detected by Faster R-CNN. From these comparisons, we conclude that exploiting effective human-related regions and designing efficient techniques to capture them are promising to improve the performance of action recognition in videos.

6. Conclusion

We have proposed a novel human-related region-based multi-stream convolutional neural network (HR-MSCNN) for action recognition. The idea is derived from the intrinsic characteristic that local motion features in the video contribute to the action label. By employing an improved version of B-RPCA (IB-RPCA), the foreground actor can be accurately detected under complex realistic situations such as noise, various illumination conditions and partial occlusions. Additionally, a simple yet effective motion saliency measure is used to efficiently extract one region that corresponds to the part of the body with the most discriminative motion. Evaluation on challenging datasets and comparisons with the state-of-the-art demonstrate that our method achieves superior action recognition performance on common benchmark datasets including UCF101 and JHMDB.

Our work can be further improved. First, the proposed IB-RPCA approach is not computationally efficient, and it can be seriously affected by background motion. It could therefore be beneficial to use more effective actor-tube detectors. Also, the moving body part that is captured by our motion saliency measure relies on the detected human body. If the detected human is incorrect, the captured acting body part will consequently also be incorrect.

Second, the assumption that a single additional HR is optimal for all action classes is unlikely. We plan to analyze the type and number of HRs that can bring the most significant contribution to action recognition. Finally, we have used a simple approach to fuse the predictions from the three regions. A method that is able to fuse them adaptively with regard to the characteristics of different streams might take advantage of complementary and redundant information in each region, and thus improve the final classification.

Acknowledgments

This work is supported by the Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114, and by grants from Office of Naval Research, US. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR. It was also supported by the Natural Science Foundation of China (61501198), Natural Science Foundation of Hubei Province (2014CFB461), Wuhan Youth Science and Technology Chenguang program (2014072704011248), the Dutch national program COMMIT and Dutch NWO TOP grant ARBITER.

References

[1] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. 28 (6) (2010) 976–990.
[2] J. Aggarwal, M. Ryoo, Human activity analysis: a review, ACM Comput. Surv. 43 (16) (2011).
[3] G. Cheng, Y. Wan, A. Saudagar, K. Namuduri, B. Buckles, Advances in human action recognition: a Survey, CoRR abs/1501.05964 (2015).
[4] E. Ijjina, K. Chalavadi, Human action recognition in RGB-D videos using motion sequence information and deep learning, Pattern Recognit. 72 (2017) 504–516.
[5] R. Qiao, L. Liu, C. Shen, A. Hengel, Learning discriminative trajectorylet detector sets for accurate skeletonbased action recognition, Pattern Recognit. 66 (2017) 202–212.
[6] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Neural Inf. Process. Syst. (2012) 1106–1114.
[7] G. Cheron, I. Laptev, C. Schmid, P-CNN: Pose-based CNN features for action recognition, Comput. Vis. Pattern Recognit. (2015) 3218–3226.
[8] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, Comput. Vis. Pattern Recognit. (2015).
[9] Z. Tu, J. Cao, Y. Li, B. Li, MSR-CNN: Applying motion salient region based descriptors for action recognition, Int. Conf. Pattern Recognit. (ICPR) (2016).
[10] S. Ma, S. Bargal, J. Zhang, L. Sigal, S. Sclaroff, Do less and achieve more: training CNNs for action recognition utilizing action images from the web, Pattern Recognit. 68 (2017) 334–348.
[11] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, Comput. Vis. Pattern Recognit. (2016).
[12] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, Neural Inf. Process. Syst. (2014).
[13] G. Gkioxari, J. Malik, Finding action tubes, Comput. Vis. Pattern Recognit. (2015).
[14] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 1154–1171.
[15] L. Zhoua, W. Li, P. Ogunbonaa, Z. Zhang, Semantic action recognition by learning a pose lexicon, Pattern Recognit. 72 (2017) 548–562.
[16] W. Chen, J. Corso, Action detection by implicit intentional motion clustering, Int. Conf. Comput. Vis. (2015).
[17] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, Comput. Vis. Pattern Recognit. (2016) 1933–1941.
[18] Z. Gao, L.F. Cheong, Y.X. Wang, Block-sparse RPCA for salient motion detection, Trans. Pattern Anal. Mach. Intell. 36 (10) (2014) 1975–1987.
[19] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human action classes from videos in the wild, CRCVTR-12-01 (2012).
[20] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, Int. Conf. Comput. Vis. (2011).
[21] H. Wang, C. Schmid, Action recognition with improved trajectories, Int. Conf. Comput. Vis. (2013) 3551–3558.
[22] L. Sun, K. Jia, D.Y. Yeung, B.E. Shi, Human action recognition using factorized spatio-temporal convolutional networks, Int. Conf. Comput. Vis. (2015).
[23] G. Guo, A. Lai, A survey on still image based human action recognition, Pattern Recognit. 47 (10) (2014) 3343–3361.
[24] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, Neural Inf. Process. Systems, 2014 (2014).
[25] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, Comput. Vis. Pattern Recognit. (2014).
[26] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, Comput. Vis. Pattern Recognit. (2014).
[27] P. Weinzaepfel, Z. Harchaoui, C. Schmid, Learning to track for spatio-temporal action localization, Comput. Vis. Pattern Recognit. (2015).
[28] B. Ni, P. Moulin, X. Yang, S. Yan, Motion part regularization: improving action recognition via trajectory group selection, Comput. Vis. Pattern Recognit. (2015).
[29] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.
[30] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Gool, Temporal segment networks: towards good practices for deep action recognition, Eur. Conf. Comput. Vis. (2016) 20–36.
[31] G. Gkioxari, R. Girshick, J. Malik, Actions and attributes from wholes and parts, Int. Conf. Comput. Vis. (2015).
[32] G. Gkioxari, R. Girshick, J. Malik, Contextual action recognition with R*CNN, Int. Conf. Comput. Vis. (2015).

[33] B. Singh, T. Marks, M. Jones, O. Tuzel, M. Shao, A multi-stream bi-directional recurrent neural network for fine-grained action detection, Comput. Vis. Pattern Recognit. (2016).

[34] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: deep networks for video classification, Comput. Vis. Pattern Recognit. (2015).

[35] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, Comput. Vis. Pattern Recognit. (2015).

[36] A. Vedaldi, K. Lenc, MatConvNet: convolutional neural networks for MATLAB, ACM Int. Conf. Multimedia (2015).

[37] Z. Tu, N. Aa, C.V. Gemeren, R.C. Veltkamp, A combined post-filtering method to improve accuracy of variational optical flow estimation, Pattern Recognit. 47 (5) (2014) 1926–1940.

[38] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, Eur. Conf. Comput. Vis. (2004).

[39] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Int. Conf. Learn. Represent. (2015).

[40] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, British Machine Vis. Conf. (2014).

[41] A. Bugeau, P. Perez, Detection and segmentation of moving objects in complex scenes, Comput. Vis. Image Understand. 113 (4) (2009) 459–476.

[42] Y. Wang, J. Song, L. Wang, O. Hilliges, L.V. Gool, Two-stream SR-CNNs for action recognition in videos, British Machine Vis. Conf. (2016).

[43] E. Candes, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.

[44] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrix, Math. Program. (2010).

[45] L. Wixson, Detecting salient motion by accumulating directionally-consistent flow, Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 774–780.

[46] M.D. Rodriguez, J. Ahmed, M. Shah, Action mach: a spatio-temporal maximum average correlation height filter for action recognition, Comput. Vis. Pattern Recognit. (2008).

[47] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, M.J. Black, Towards understanding action recognition, Int. Conf. Comput. Vis. (2013).

[48] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, Neural Inf. Process. Syst. (2015).

[49] V. Kantorov, I. Laptev, Efficient feature extraction, encoding, and classification for action recognition, Comput. Vis. Pattern Recognit. (2014).

[50] Z. Tu, R. Poppe, R.C. Veltkamp, Weighted local intensity fusion method for variational optical flow estimation, Pattern Recognit. 50 (2016) 223–232.

[51] J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. Schmid, EpicFlow: edge-preserving interpolation of correspondences for optical flow, Comput. Vis. Pattern Recognit. (2015).

[52] D. Butler, J. Wul, G. Stanley, M. Black, A naturalistic open source movie for optical flow evaluation, Eur. Conf. Comput. Vis. (2012).

[53] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, R. Szeliski, A database and evaluation methodology for optical flow, Int. J. Comput. Vis. 92 (1) (2011) 1–31.

[54] H. Wang, A. Klaser, C. Schmid, C. Liu, Action recognition by dense trajectories, Comput. Vis. Pattern Recognit. (2011).

[55] X. Yang, P. Molchanov, J. Kautz, Multilayer and multimodal fusion of deep neural networks for video classification, ACM Multimedia (2016) 978–987.

[56] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, Dynamic image networks for action recognition, Comput. Vis. Pattern Recognit. (2016).

[57] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, Int. Conf. Machine Learn. (2015).

Zhigang Tu started his MPhil Ph.D. in image processing at the School of Electronic Information, Wuhan University, China, 2008. He received a Ph.D. in Communication and Information System from Wuhan University, 2013. In 2015, he received the Ph.D. degree in Computer Science from Utrecht University, Netherlands. From 2015 to 2016, he was a postdoctoral researcher at Arizona State University, US. He is currently a postdoctoral research fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include motion estimation, object segmentation, object tracking, action recognition, and anomaly detection.

Wei Xie is an associate professor at Computer School of Central China Normal University, China. His research interests include motion estimation superresolution reconstruction, image fusion and image enhancement. He received his B.Eng in electronic information engineering and Ph.D. in communication and information system from Wuhan University, China, in 2004 and 2010, respectively. Then, from 2010 to 2013, he served as an assistant professor at Computer School of Wuhan University, China.

Ronald Poppe received a Ph.D. in Computer Science from the University of Twente, the Netherlands. He was a visiting researcher at the Delft University of Technology, Stanford University and University of Lancaster, respectively. He is currently an assistant professor at the Information and Computing Sciences department of Utrecht University. His research interests include the analysis of human behavior from videos and other sensors, the understanding and modeling of human (communicative) behavior and the applications of both in real-life settings. In 2012 and 2013, he received the most cited paper award from the “Image and Vision Computing” journal, published by Elsevier.

Remco C. Veltkamp is full professor of Multimedia at Utrecht University, Netherlands. His research interests are the analysis, recognition and retrieval of, and interaction with, music, images, and 3D objects and scenes, in particular the algorithm and experimentation aspects. He has written over 150 refereed papers in reviewed journals and conferences, and supervised 15 Ph.D. theses. He was director of the national project GATE - Game Research for Training and Entertainment.

Baoxin Li received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2000. He is currently a full professor and Chair of computer science and engineering with Arizona State University, US. From 2000 to 2004, he was a senior researcher with SHARP Laboratories of America, Camas, WA, where he was the technical lead in developing SHARPs HiIMPACT Sports technologies. From 2003 to 2004, he was also an adjunct professor with the Portland State University, Portland, OR. He holds nine issued US patents. His current research interests include computer vision and pattern recognition, image/video processing, multimedia, medical image processing, and statistical methods in visual computing. He won the SHARP Laboratories President Award twice, in 2001 and 2004. He also received the SHARP Laboratories Inventor of the Year Award in 2002. He received the National Science Foundations CAREER Award from 2008 to 2009. He is a senior member of the IEEE.

Junsong Yuan (M’08–SM’14) received his Ph.D. from Northwestern University and M.Eng. from National University of Singapore. Before that, he graduated from the Special Class for the Gifted Young of Huazhong University of Science and Technology, Wuhan, China, in 2002.

He is currently an associate professor at Computer Science and Engineering department of State University of New York at Buffalo. Before that, he was an associate professor at Nanyang Technological University (NTU), Singapore. His research interests include computer vision, video analytics, gesture and action analysis, large-scale visual search and mining. He received best paper award from Intl. Conf. on Advanced Robotics (ICAR’17), 2016 Best Paper Award from IEEE Trans. on Multimedia, Doctoral Spotlight Award from IEEE Conf. on Computer Vision and Pattern Recognition (CVPR’09), Nanyang Assistant Professorship from NTU, and Outstanding EECS Ph.D. Thesis award from Northwestern University.

He is currently Senior Area Editor of Journal of Visual Communications and Image Representations (JVCI), Associate Editor of IEEE Trans. on Image Processing (T-IP), IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT), and served as Guest Editor of International Journal of Computer Vision (IJCV). He is Program Co-chair of ICME’18 and VCIP’15, and Area Chair of ACM MM’18, ICPR’18, CVPR’17, ICIP’18’17, ACCV’18’14 etc.