Pattern Recognition 103 (2020) 107312

Contents lists available at ScienceDirect

Pattern Recognition

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

Learning motion representation for real-time spatio-temporal action localization

Dejun Zhang[a,*], Linchao He[b], Zhigang Tu[c,*], Shifu Zhang[d], Fei Han[b], Boxiong Yang[e]

[a] School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China
[b] College of Information and Engineering, Sichuan Agricultural University, Yaan 625014, China
[c] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University, Wuhan, 430079, China
[d] Shenzhen Infinova Ltd. Company, Infinova Building, Guanlan Hi-tech Industrial Park, Shenzhen 518100, China
[e] School of Information and Intelligence Engineering, University of Sanya, Sanya 572022, China

ARTICLE INFO

Article history:
Received 20 July 2019
Revised 17 February 2020
Accepted 24 February 2020
Available online 27 February 2020

Keywords:
Spatio-Temporal Action Localization
Real-time Computation
Optical Flow Sub-network
Pyramid Hierarchical Fusion

ABSTRACT

The current deep learning based spatio-temporal action localization methods that using motion information (predominated is optical flow) obtain the state-of-the-art performance. However, since the optical flow is pre-computed, leading to these methods face two problems – the computational efficiency is low and the whole network is not end-to-end trainable. We propose a novel spatio-temporal action localization approach with an integrated optical flow sub-network to address these two issues. Specifically, our designed flow subnet can estimate optical flow efficiently and accurately by using multiple consecutive RGB frames rather than two adjacent frames in a deep network, simultaneously, action localization is implemented in the same network interactive with flow computation end-to-end. To faster the speed, we exploit a neural network based feature fusion method in a pyramid hierarchical manner. It fuses spatial and temporal features at different granularities via combination function (*i.e.* concatenation) and point-wise convolution to obtain multiscale spatio-temporal action features. Experimental results on three publicly available datasets, e.g. UCF101-24, JHMDB and AVA show that with both RGB appearance and optical flow cues, the proposed method gets the state-of-the-art performance in both efficiency and accuracy. Noticeably, it gets a significant improvement on efficiency. Compared to the currently most efficient method, it is 1.9 times faster in the running speed and 1.3% video-mAP more accurate on the UCF101-24. Our proposed method reaches real-time computation for the first time (up to 38 FPS).

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Current spatio-temporal action localization methods focus on processing untrimmed, multi-target videos rely on the popular object detection frameworks [1]. Usually, these localization methods perform classification on each action class and aim to detect the duration of every single action instance. The human action is firstly detected at the frame level, and then a dynamic linking algorithm is used to transform a series of frame detections to generate human action tubes. We can perform spatio-temporal action localization based on the generated action tubes. Compared to the traditional work (*i.e.* dense trajectories) [2], Convolutional Neural Network (CNN) based methods [3–5] get better performance in both accuracy and efficiency. Optical flow [6–8] is widely used in deep learning frameworks to help handle the task of action localization in videos [3,9], however, none of them perform the task of localization completely in an end-to-end way since they only utilize the pre-computed optical flow as a part of network input.

The recent CNN-based optical flow methods can achieve real-time speed with GPU acceleration (*i.e.* FlowNet [10], FlowNet 2.0 [11]), but integrating them into an entire network to jointly estimate optical flow and localize human actions end-to-end is still a hard problem. In this work, *we firstly argue that a real-time action localization model should be able to localize human actions spatio-temporally without pre-computing optical flow. Secondly, optical-flow-required motion features and action-like features should be learned and fine-tuned jointly in one network to improve the performance of each other interactively.*

To address the first issue, we exploit a novel spatio-temporal action localization model, which combines the localization network with a CNN based optical flow estimation subnet, focuses on how to generate good detections. The optical flow subnet can learn effective optical-flow-required features and estimate accurate optical flow. Besides, the same backbone network is used to extract both

* Corresponding authors.
E-mail addresses: zhangdejun@cug.edu.cn (D. Zhang), tuzhigang@whu.edu.cn (Z. Tu).

https://doi.org/10.1016/j.patcog.2020.107312
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

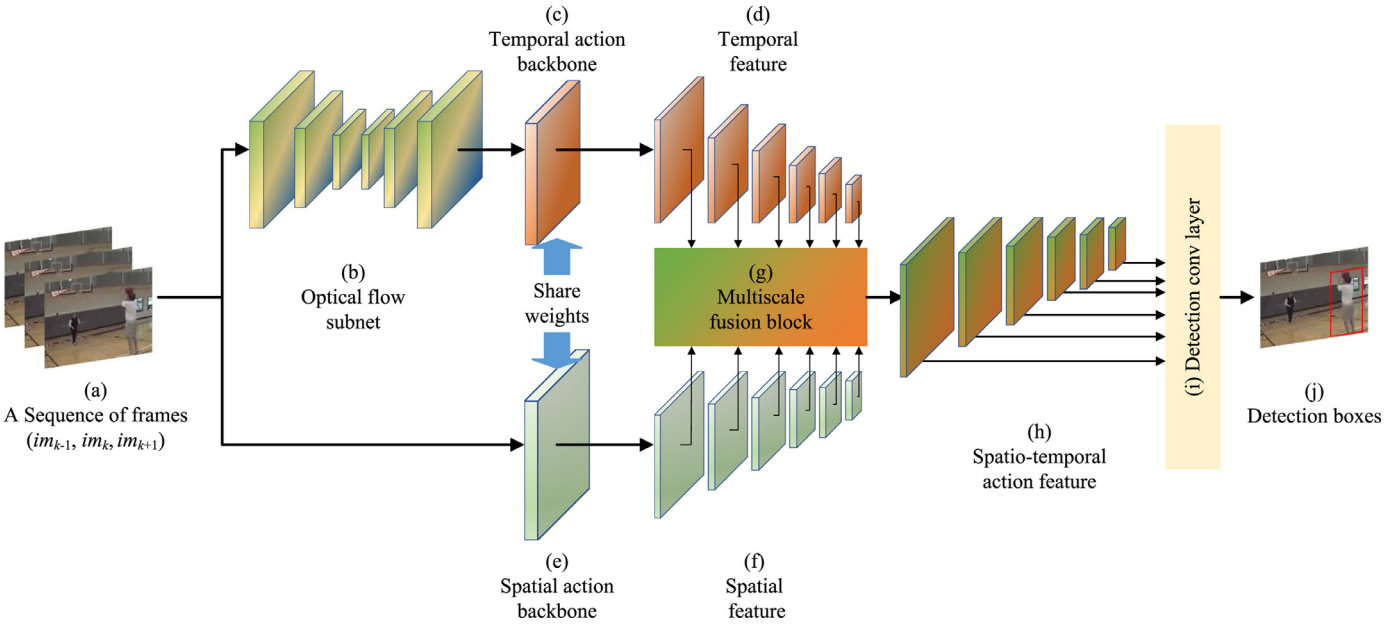

Fig. 1. The input of our model is a sequence of RGB frames (**a**). In the optical flow path, frames are split into consecutive frame tuples (*i.e.* (im_{k-1}, im_k) and (im_k, im_{k+1})) which are taken as input by optical flow subnet (**b**). The temporal action detector (**c**) generates multi-scale temporal features (**d**) from the output of optical flow subnet. The spatial action detector is conducted on frame im_k and outputs multi-scale spatial features. We use a fusion block (**g**) to combine temporal and spatial features to produce multi-scale spatio-temporal action features (**h**). Finally, a detection convolution layer is used to detect the action bounding boxes from the fused action features.

motion and RGB appearance deep features from the computed optical flow and the original video frames (see Fig. 1) for the goal of action localization. An one-stage detector [12,13], which can detect both motion and RGB features in real-time (about 37.8 FPS), is introduced.

To handle the second issue, we focus on *how to interactively fuse the deep features of RGB appearance and motion in the learning process effectively*. For a video task, motion is an essential element and usually plays a more important role than RGB appearance. Nowadays, most of the top-ranking action recognition methods adapt a two-stream architecture [14] and fuse their spatial-stream and temporal-stream features for action classification. The spatio-temporal action localization approaches [3,9,15] usually apply late fusion scheme (*e.g.* union-set fusion and element-wise add fusion) to combine bounding boxes of the two streams. The late fusion is problematic as it ignores low-level features while only focuses on high-level features leading to incorrect object localization. To improve the drawback, inspired by SSD [16], we apply the multiscale prediction scheme to fuse motion and appearance features at different scales. We propose a fusion block in a neural network fashion, where a combination function is used to merge spatial and temporal features and deploy 1×1 convolution to make spatial and temporal information interacting with each other to generate spatio-temporal action features (Fig. 4**a**). However, the above method still lacks interaction between different scales, thus we upsample the previous small size spatio-temporal features to match their corresponding high-level information, and use point-wise convolution to adjust the number of channels to fit the current feature maps. Finally, the spatial, temporal and upsampled features are combined by addition.

Our main contributions can be summarized as follows:

- We propose a novel method to localize human actions in videos spatio-temporally with integrating an optical flow estimation subnet. The designed new architecture can perform action localization and optical flow estimation jointly in an end-to-end manner. The interaction between the action detector and flow subnet enables the detector to learn parameters from appearance and motion simultaneously and guiding flow subnet to compute task-specific optical flow.
- We exploit an effective fusion method to fuse appearance and optical flow deep features in a multi-scale fashion. It captures semantic information in both coarse and fine levels, which are useful for producing more accurate predictions. Besides, the multi-scale temporal and spatial features are combined interactively to model a more discriminative spatio-temporal action representation.
- The presented method achieves real-time computation at the first time with the usage of both RGB appearance and optical flow. Compared to the representative efficient [3] and inefficient [17] methods, our approach achieves 1.9 times and 9.9 times faster respectively. Additionally, our method also outperforms the state-of-the-art method [3], which gets the highest efficiency to localize human action, by 1.3% in accuracy on the UCF101-24 dataset.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. Section 3 outlines the overall framework of our network for real-time spatio-temporal action localization and presents some implementation details, such as integrated optical flow sub-network, multiscale action localization network and detection Convolution Layer. In Section 4, we introduce the details of our experimental evaluation setup and present the experimental results on two public benchmarks. Finally, the conclusions and future work are discussed in Section 5.

2. Related work

Recently, most spatio-temporal action localization methods [3,9,18] are based on CNN object detectors [16,19]. The classical optical flow approaches [20,21], which are used to localize human actions, are hard to be combined with action detectors and greatly increase the running cost. In the following, we briefly review the recent CNN-based object detectors, optical flow methods, and spatio-temporal action localization approaches.

2.1. CNN-based detectors

CNNs have been successfully used for detecting objects [12,19,22]. R-CNN [23] and its extended network [19] consider object detection as a region propose generation and classification problem. Faster R-CNN [19] obtains a good performance on accuracy by using a Region Proposal Network (RPN), but it can only process about 6 frames per second, caused by a heavy head network to predict the bounding boxes and the confidence scores. To accelerate object detection speed, the one-stage detector is introduced by YOLO [13], which uses fixed anchor boxes to generate regions instead of selective search [23] and reaches, to predict 45 frames per second. SSD [16] uses pyramidal features to overcome the drawback of one-stage detectors which cannot locate small targets. Inspired by the anchor boxes of Faster R-CNN, SSD proposes similar prior boxes with multiple sizes and aspect ratios to make it is able to fit object position with more complex shapes. SSD gets better accuracy with a still real-time detection speed.

2.2. Optical flow networks

CNN-based optical flow method is becoming increasingly popular [8,24], which treats the optical flow estimation as a neural network optimization problem. FlowNet [10] is the first work to use CNN to directly predict optical flow from RGB frames and models the prediction as a supervised deep learning problem. But the FlowNet has a weakness in the accuracy compared to the traditional optical flow methods. FlowNet2 [11] significantly boosts the accuracy and obtains comparable results in contrast to traditional methods, while only costs a small price in speed. FlowNet related networks require a large number of labeled ground truth to train the parameters, however, it is hard to get the ground truth data in real applications. Fan [24] integrates the classical TV-L1 [21] into a neural network to form a TVNet which can perform a basic optical flow estimation efficiently without the requirement of training parameters. Most recently, a PWC-Net [25] is exploited according to several good practical principles: pyramidal processing, warping, and cost volume processing. Compared to FlowNet2, PWC-Net has a smaller size, faster speed, and more accurate results. Importantly, it is can be integrated into another task-specific neural network. We utilize the PWC-Net in our integrated optical flow sub-network to balance efficiency and accuracy.

2.3. Action localization

The sliding window is initially used in spatio-temporal action localization [26]. Gemert et al. [27] show dense-trajectories have a powerful ability to perform spatio-temporal action localization, but it fails in detecting small motion. RPN is introduced for action localization by Saha et al. [17], Peng and Schmid [28] to generate action proposal on frame-level, and dynamic programming is adopted to predict the video-level label. However, the efficiency of two-stage detectors is low. One-stage detector is introduced by Gurkirt et al. [3] to perform a real-time detection. ROAD [3] produces action bounding boxes for both appearance and flow frames and uses an online algorithm to incrementally construct and label action tubes from boxes. But its speed is very low (7 fps) if conducting detection on both motion and RGB appearance. Kalogeiton et al. [18] propose an Action Tubelet detector (ACT-detector) in which the input is a sequence of frames and the output is the tubelets. ACT also processes optical flow and RGB appearance separately, and cannot detect human actions in real-time. In contrast, we propose a novel model, which combines the appearance and the optical flow inside one neural network, to predict action bounding boxes in real-time.

3. Joint action localization and motion estimation

As shown in Fig. 1, we propose an end-to-end action detection network to predict detection boxes and their class-specific scores for multiple RGB frames input $\{im_{k-1}, im_k, im_{k+1}\}, im \in \mathbb{R}^{W\times H\times 3}$ where W and H respectively denote the width and height of the im.

3.1. Integrated optical flow sub-network

We design a modified two-stream deep network to process the visual cues of motion and appearance with shared weights as [3,15]. The typical two-stream CNNs require the input of both RGB frames and optical flow, where the optical flow is precomputed costly from the source video images, in contrast, the input of our magic network is only a sequence of RGB images (Fig. 1a). To faster the localization of human actions spatio-temporally, we integrate the optical flow estimation network to our network (Fig. 1b).

We utilize PWC-Net [25], which contains several good practices for optical flow estimation, e.g., image pyramid, warping, and cost volume, as the baseline of our optical flow sub-network. Let Θ be a set of all the trainable parameters, f^i_Θ denotes the flow field at the ith pyramid level, and f^i_{GT} represents the corresponding ground truth flow field. The training loss is defined as:

$$L(\Theta) = \sum_{i=i_0}^{I} \alpha_i \sum_x |f^i_\Theta(x) - f^i_{GT}(x)|_2 + \gamma|\Theta|_2 \tag{1}$$

where $|\cdot|_2$ computes the L2 normalization of a vector, x is the pixel index, and $\gamma|\Theta|_2$ regularizes parameters.

We modify the PWC-Net to enable it to process a sequence of frames to generate flow frames. We stack every two adjacent frames (*i.e.* $\{im_{k-1}, im_k\}, \{im_k, im_{k+1}\}$) at the channel axis to get two $3D$ tensors F_1 and F_2, where each has a dimension of $W \times H \times 6$. The tensor size needs to be rescaled to a multiple of 64. We rescale $W \times H$ to 320×320 to form new tensors $F_1^{'}$ and $F_2^{'}$, which are used to learn optical-flow-like features O_1 and O_2. Flow subnet $\mathcal{P}$ takes two adjacent frames $\{im_a, im_b\}$ as input to yield the flow features f with parameters Θ:

$$f = \mathcal{P}(\{im_a, im_b\}, \Theta) \tag{2}$$

Multiple image pairs increase the model complexity, and have a little promotion on short-term temporal information. Therefore, we use one pair of images to estimate optical flow information to consider the balance between complexity and precision.

3.2. Fusion multiple optical frames

As shown in Fig. 1, the PWC-Net is employed to estimate optical flow without modifying its architecture. We need to fuse the short-term temporal information which is generated by the PWC-Net from multiple optical flow frames. A convolutional layer is added to fuse the temporal information, as shown in Fig. 2. The fused temporal information has an identical shape with the RGB frame. We can reuse the weights of the backbone networks to detect action instances. By using the flow subnetwork, we can achieve comparable results in real-time on a single GPU.

3.3. Multiscale action localization network

We follow the footstep of the one-stage object detector [13,16] to predict bounding boxes and confidence scores simultaneously without using RPN. We use the shared weights network – Deeplab-VGG16 to learn spatial and temporal deep features and fuse them in multiscale in fusion blocks to form spatio-temporal

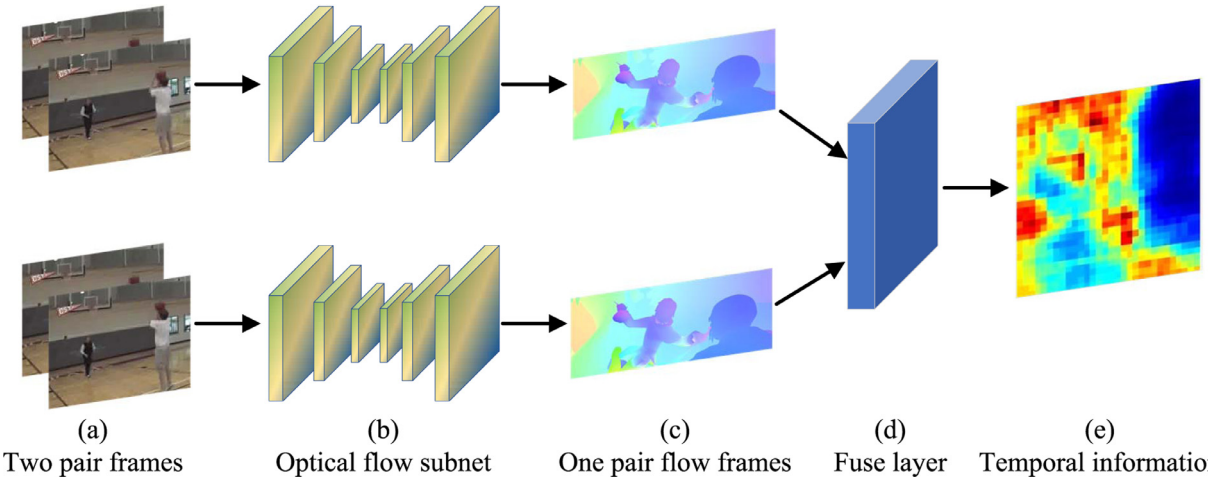

Fig. 2. Our optical flow subnet (**b**) takes two pair RGB frames (**a**) as input and generates one pair flow frames (**c**). In the fuse convolutional layer (**d**), we use convolution to change the shape of flow frames from $W \times H \times 6$ to $W \times H \times 3$. Therefore, the flow frames is encoded as fused temporal information which can be processed by the backbone network.

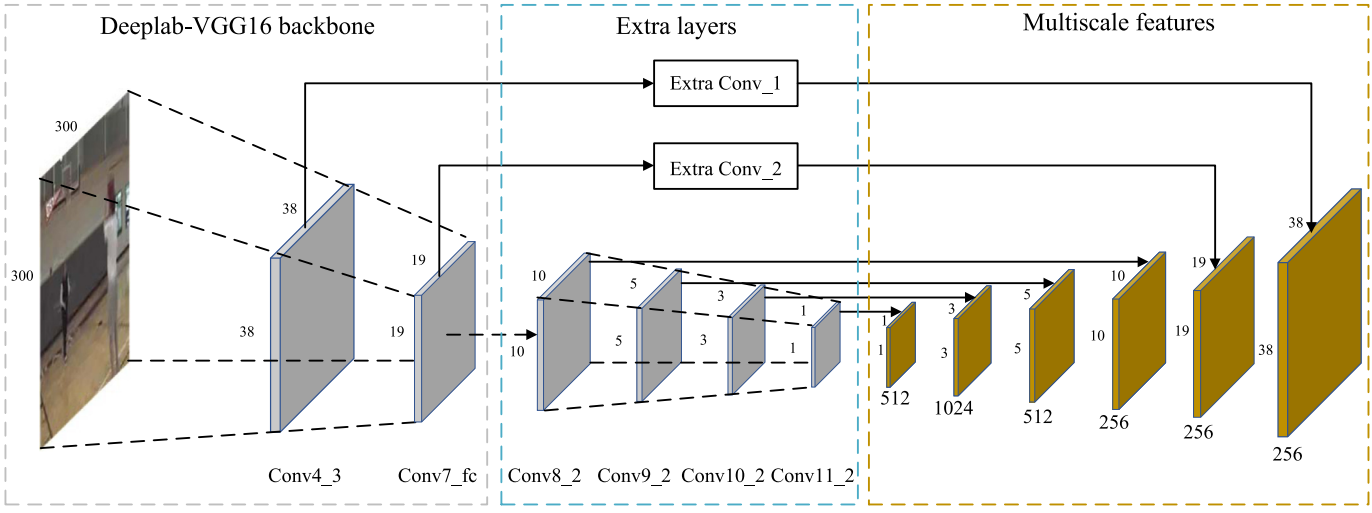

Fig. 3. The multiscale feature maps are generated from the Deeplab-VGG16 backbone [29] and the extra layers. We use the shared weights backbone and the extra layers to compute features for the spatial stream and temporal stream. From the first to the sixth levels, the number of feature channels are respectively {512, 1024, 512, 256, 256, 256}.

feature maps. The feature maps contain detailed hierarchical information, which is useful for the convolutional detection layer to better predict the bounding boxes and their class-specific confidence scores.

3.3.1. Spatial path backbone

The spatial path backbone (Fig. 1e) computes appearance convolutional features from im_k. We preprocess im_k to meet the requirement of the spatial backbone (*i.e.* resize and mean subtraction). After the forward propagation of Deeplab-VGG16 backbone network [29] and the extra layers (see in Fig. 3), we can obtain a sequence of multiscale appearance feature maps $\{m_1^a, ..., m_K^a\}$ (Fig. 1f). From the first to the sixth levels, the number of feature channels are respectively {512, 1024, 512, 256, 256, 256}. The spatial action detector models the appearance information as follows:

$$M^a = \mathcal{F}(im_k, \mathbf{W}_b) \tag{3}$$

where $\{m_1^a, ..., m_K^a\} \in M^a$. $\mathcal{F}(\cdot, \cdot)$ is a function represents a ConvNet with parameters $\mathbf{W}_b$ which operates on frame im_k to produce multiscale spatial features M^a.

3.3.2. Temporal path backbone

We use bilinear upsampling to resacle the size of $\{O_1, O_2\}$ to $W \times H$. To reuse the neural network parameters from spatial action detector, we stack $\{O_1, O_2\}$ at channel and perform a 1×1 pointwise convolution on above tensor O. We feed O into the temporal action detector (Fig. 1c), which shares weights with spatial action backbone, to learn the multiscale optical flow feature maps $\{m_1^{of}, ..., m_K^{of}\}$ (Fig. 1d). The temporal action detector models the optical-flow-like information as follows:

$$M^{of} = \mathcal{F}(f, \mathbf{W}_b) \tag{4}$$

where $\{m_1^{of}, ..., m_K^{of}\} \in M^{of}$. $\mathcal{F}(\cdot, \cdot)$ is a ConvNet function with parameters $\mathbf{W}_b$, which is used to operate on flow feature map f to extract multiscale temporal features M^{of}.

3.3.3. Multiscale feature fusion block

The multiscale feature fusion block is an important component in our spatio-temporal action detector. Previous works [3,18] normally fuse appearance and optical flow action bounding boxes (tubelets) by using union fusion and late fusion. However, they only combine the detected boxes at frame level rather than fus-

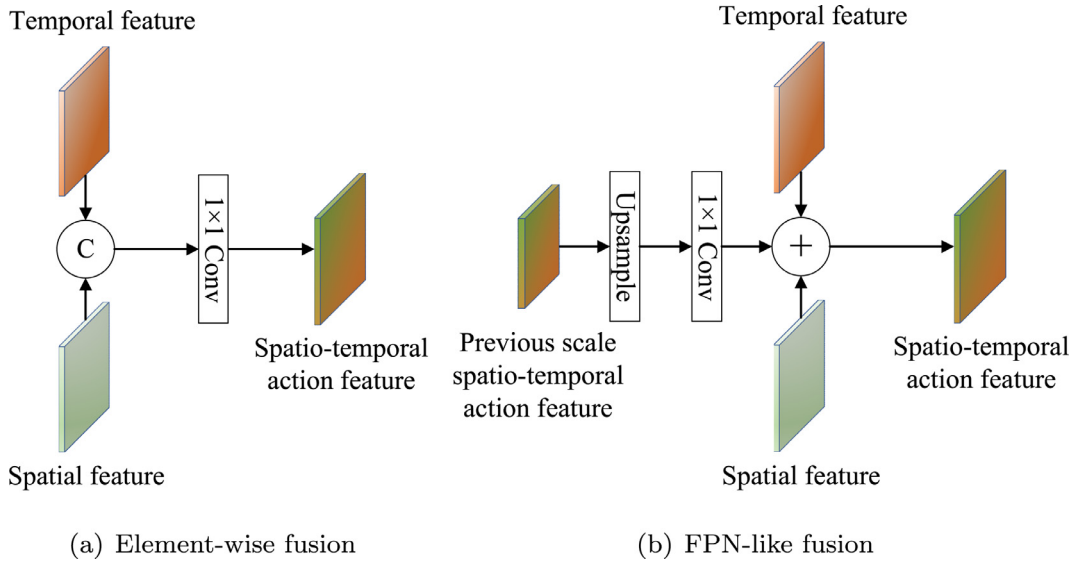

Fig. 4. (**a**) Element-wise fusion: we use the combination function (concatenate, sum and average) to combine the temporal and spatial deep features. 1 × 1 Convolution is deployed to exchange information between channels, thus we can obtain a multiscale spatio-temporal action feature. (**b**) FPN-like fusion: we upsample the spatio-temporal action feature map from a previous small scale to match the size. 1 × 1 convolution is used to match the channel of the current spatio-temporal action feature. According to FPN [30], we combine these feature maps by addition to generate a spatio-temporal action feature representation.

ing the extracted features. Instead, we present an effective neural network-based feature aggregation block $\mathcal{G}$ to fuse spatial and temporal features in a pyramid multiscale manner (Fig. 4). We can implement the fusion block in two ways: element-wise fusion (Fig. 4a) and pyramid fusion (Fig. 4b). Element-wise fusion uses the element-wise function (*i.e.* average) to combine tensors. The neural network is introduced into element-wise fusion to interact between channels. Pyramid fusion utilizes the small scale feature maps to enhance the semantic information of finer scale feature maps.

Element-wise fusion. Element-wise fusion contains two parts: *combination function* and *fusion function*. *Combination function* combines spatial and temporal features by element-wise computation. *Fusion function* uses a neural network to make features interacting between channels.

$$\mathcal{G}\left(M^a, M^{of}\right) = \sum_{i=1}^{K} S\left(C\left(m_i^a, m_i^{of}\right), \mathbf{W}_p\right) \tag{5}$$

$C(\cdot,\cdot)$ denotes a combination function that aims to incorporate two tensors into one. $S(\cdot, \mathbf{W}_p)$ represents a neural network with parameters $\mathbf{W}_p$ to fuse spatial and temporal features.

Combination function. We test three types of combination functions: concatenate, sum and average. 1) **Concatenate**: It can reserve all the information from both appearance and optical flow streams, leading the neural network to learn better representation. 2) **Sum**: By deploying an element-wise addition function to fuse spatial and temporal features, and getting the superposition of discriminative features from both streams. 3) **Average**: one alternative way to add function which merges tensors to balance the response from two streams.

Fusion function. We can obtain $\hat{m}_i$ by using different types of combination function C, but $\hat{m}_i$ lacks the interaction between channels, resulting in appearance and motion information cannot be well exchanged. To handle this problem, in this work, the element-wise convolution is introduced to fuse features between channels and reduce channel numbers (*i.e.* concatenate function):

$$m = S\left(\hat{m}, \mathbf{W}_p\right) \tag{6}$$

Where m denotes the set of multiscale spatio-temporal action features, i.e., $\{\hat{m}_1, ..., \hat{m}_k\} \in \hat{m}$.

Spatio-temporal feature pyramid fusion. Element-wise fusion makes feature interacting between spatial and temporal paths, but it lacks the interaction between different scales. Inspired by FPN [30], we use a high-level feature map to enhance the semantic information of a low-level feature map, making features interact between scales. The multiscale block, which called the Spatio-temporal feature pyramid (STFP) fusion, involves a bottom-up pathway, a top-down pathway, and lateral connections.

Bottom-up pathway. The bottom-up pathway uses feed-forward backbone CNN to compute features. As we aim to perform prediction at multi-scales, we use different stage outputs of the backbone CNN which are grouped by scales $\{m_1^a, ..., m_K^a\}$ and $\{m_1^{of}, ..., m_K^{of}\}$. For the two-stream network, we build two shared weights bottom-up pathway p^a and p^{of}, which are dubbed RGB pathway and optical flow pathway, to extract features.

Top-down pathway. The top-down pathway upsamples the low-resolution and strong semantic feature map to next finer-scale (*i.e.* 19 × 19 → 38 × 38). An element-wise convolution is introduced to adjust the channel of the former spatio-temporal feature map m_{i-1}.

Lateral connections. We build the block to construct our feature maps with lateral connections. We merge the upsampled spatio-temporal feature map m_{i-1} with RGB pathway feature m_i^a and optical flow pathway m_i^{of} by element-wise addition. The designed block is used to combine high-level and high-resolution semantic information, in this way, our model becomes more robust and can detect small objects more accurately.

$$\mathcal{G}\left(M^a, M^{of}, W_p\right) = \sum_{i=2}^{K} U(m_i) + \sum_{i=1}^{K} \left(m_i^a + m_i^{of}\right) \tag{7}$$

Where $U(\cdot)$ denotes an adjustment function that uses upsample and point-wise convolution to adjust the size of spatio-temporal feature maps with parameter W_p.

3.4. Detection Convolution Layer

Due to the good performance of SSD [16] and RFB Net [12], we use the same cascade prediction layers as them (Fig. 1i). Differently, the input RGB feature maps are replaced by our spatio-temporal action features m. The detection layer outputs the action

classes and the coordinate offsets (Fig. 1j), thus every feature map is $D_i \in \mathbb{R}^{p\times l\times b\times(c+4)}$, where p and l denote width and height of D_i, b is the default boxes of the RFB Net, and $c+4$ represents the action classes c and the coordinate offsets.

3.5. Training loss

We modify the loss function of RFB Net by adding the parameter of the integrated optical-flow sub-network, shown as follows:

$$L = L_{conf} + L_{reg} + L(\Theta) \tag{8}$$

where L_{conf} and L_{reg} are the class confidence loss function [16] and the 2D bounding box coordinate regression loss function respectively. $L(\Theta)$ is the integrated optical-flow sub-network loss function with parameter Θ, defined in Eq. (1). It should be noted that we freeze the parameters of optical-flow sub-network at the beginning of training.

4. Experiments

In this section, extensive experiments are conducted to test the proposed method whether it is able to effectively localize human actions spatio-temporally in real-time on the widely used untrimmed video dataset UCF101-24 [31], a fully annotated dataset JHMDB [32], and the atomic action dataset AVA [15]: 1) evaluating the performance of the exploited multiscale fusion strategy; 2) comparing the accuracy of our method with the state-of-the-arts; 3) analyzing the running speed of the whole network with joint tasks. The code is publicly released in our community site[1].

4.1. Dataset

UCF101-24, which is a subset of UCF101 [31] with spatio-temporal labels, is one of the largest and most challenging action datasets for spatio-temporal action localization. Each video may contain multiple action instances with same action class. UCF101-24 is composed of 24 classes with 3207 videos which is corrected by Singh et al. [3]. Following with previous works [3,9,18], we report the experiment results on the first split.

JHMDB [32] contains 928 videos with 21 actions. All the action videos are trimmed. We report the frame-mAP results averaged on the three splits.

AVA [15] is a video dataset of atomic visual action which contains 80 action classes sampling at 1 fps. Following the evaluation protocol [15], we report the most frequent 60 classes and the frame-level mAP.

4.2. Metrics

We use the intersection-over-union (IoU) metric at the frame and video level to evaluate the performance of spatial and temporal localization. For frame level, we follow the standard protocol used by PASCAL VOC object classes challenge [33] and report the average precision (AP) using an IoU threshold of 0.5. For the whole dataset, we report the mean average precision (mAP) over all classes. For video level, we compute the spatio-temporal IoU between the ground truth tube and linked detection tubes (the linking method is defined in [3]) with the threshold of 0.5, we also report the mAP (video) over all classes.

[1] https://github.com/djzgroup/RT-ST-Action-Localization

Table 1
Comparison (F-mAP) of different flow subnetworks on UCF-101-24. (The IoU threshold is set to 0.5).

Flow Subnetwork	F-mAP@0.5	Time(millisecond)
PWC-Net [25]	**67.7**	6
FlowNet2 [11]	58.6	60
FlowNet2S [11]	38.7	**2**
Brox Flow [20]	56.8	110

4.3. Implementation details

We use the ImageNet pretrained model to initialize the backbone network of RFB Net [12]. The integrated flow subnet uses the weights offered by Sun et al. [25]. We perform data augmentation to the frames (both flow and appearance streams with the same settings). Specially, we use photometric distort, random crop, random mirror, and channel swapping. The implementation is carried on Pytorch. We train our model on a 4-GPU machine and each GPU with 11 GB VRAM has 10 sequences in a mini-batch (so in total with a mini-batch size of 40 sequences). We compute the gradients from all GPUs and perform backpropagation on the main GPU, the others copy the weights from the main GPU. The learning rate of SGD with momentum is set to 0.001 and decreases by 0.1 at 30k, 60k and 100k iterations. The momentum and weight decay rate of SGD is set to 0.9 and 0.0005 respectively. We stop the training after 150k iterations. The model is trained in an end-to-end fashion, and we first freeze the weights of the integrated optical flow network and then unfreeze weights after the loss stable. The length K of input RGB frames is 3.

4.4. Optical Flow Subnetworks Comparison

In this subsection, we evaluate four kinds of optical flow subnetworks, i.e., PWC-Net [25], FlowNet2 [11], FlowNet2S [11], and Brox Flow [20]. To be fair, we input the images with the same resolution (320×320) to the flow subnetwork and set the parameters of flow subnetworks with pretrained weights. We report the frame-mAP (F-mAP) on UCF-101-24 dataset in Table 1.

From Table 1, we can find that the neural network based flow subnetworks (*i.e.* PWC-Net, FlowNet2) are faster than the traditional optical flow algorithms (*i.e.* Brox Flow). FlowNet2 has almost the same accuracy as Brox Flow [20] while costs much less time. FlowNet2S only needs 2ms to compute an optical flow, but its result is the worst. PWC-Net gets the highest F-mAP and costs 6ms to estimate optical flow. Consequently, we choose the PWC-Net as the flow subnetwork in our method.

4.5. Multiscale fusion method study

In this subsection, we focus on the study of multiscale fusion method. For fair comparison, we unfreeze the flow subnet at 75k iterations and stop training at 150k iterations. We evaluate four multiscale fusion methods: (1) *M-Concat.* (i.e.*M-Concatenate*), (2) *M-Sum*, (3) *M-Average*, and (4) *STFP. M-Concat.*, *M-Sum* and *M-Average* belong to element-wise fusion reported on UCF-101-24. The experimental results are shown in Table 2.

Firstly, from the bottom part of Table 2, we can find that *M-Concat.* fusion achieves the best performance. Since UCF101-24 includes many big targets, we can infer that the interaction between spatial and temporal deep features is more important than the interaction between scales. In particular, the interaction between spatial and temporal information is very useful to improve the performance of action prediction in case that the prediction heavily relied on temporal information. The interaction between scales mainly boosts the detection performance of small targets which is

Table 2
Comparison of different fusion blocks on UCF-101-24. (The IoU threshold is set to 0.5).

Method	Fusion	Multiscale	F-mAP@0.5	V-mAP@0.2
M-Concat.	✓		37.8	39.1
M-Concat.		✓	11.7	18.9
M-Concat.	✓	✓	**67.7**	**74.8**
M-Sum	✓	✓	55.9	64.1
M-Average	✓	✓	58.9	67.9
STFP	✓	✓	56.3	66.0

Table 3
Comparison (frame-mAP) to the state-of-the-arts on the UCF-101-24 dataset. (The IoU threshold is set to 0.5).

Method	Detector	mAP@0.5
Weinzaepfel et al. [34]	R-CNN	35.8
Peng et al. (w/o MR) [28]	Faster R-CNN	64.8
Peng et al. (w/ MR) [28]	Faster R-CNN	65.7
Hou et al. [35]	Tube Proposal	41.4
Ours	RFB Net	**67.7**

not the majority in UCF101-24. *M-Concat.* can maximize the retention of spatial and temporal features due to it concatenates tensors into one. While other fusion methods, which use element-wise addition or average function, blur the spatial and temporal information and leading to bad performance.

Secondly, the performance of the fusion method and the non-fusion method is compared. As shown in Table 2, with the usage of any one of the four multiscale fusion approaches, the accuracy is improved by at least 44.2% (55.9% vs 11.7%) and 45.2% (64.1% vs 18.9%) on F-mAP and video mAP (V-mAP) respectively. We can infer that our detection network is unable to learn good representations if we don't fuse spatial and temporal features. The results prove that our fusion strategy greatly improves the confidence score and IoU overlaps of the detection boxes.

Thirdly, we compare the performance of the fusion method using or without using the multiscale strategy. Under the same fusion strategy, our multiscale strategy improves the frame and video mAP by 29.9% (67.7% vs 37.8%) and 35.7% (74.8% vs 39.1%) respectively. The results demonstrate that the multiscale strategy offers the more discriminate features to generate detection boxes.

Accordingly, we choose the *M-Concat.* fusion with a multiscale strategy as our default setting due to it can well balance speed and accuracy.

4.6. Comparison with state-of-the-art

We compare the proposed approach with several state-of-the-art methods. It should be noted that previous works reported in this subsection use RGB and optical flow to perform spatio-temporal action localization if not specified.

4.6.1. Frame-mAP

We report the UCF-101-24 frame-mAP results in Table 3. The R-CNN [23] based method [34], the Faster R-CNN [19] based approaches [28], and the tube proposal scheme [35] are selected for comparison. For [28], we report the experimental results with and without the multi-region method. Our approach achieves 67.74% mAP with the IoU threshold of 0.5, and outperforms [28,34,35] for at least 31.9% due to our unified detection network uses the integrated flow subnet and multiscale fusion block. [28,34,35] lack the capacity of fusing spatial and temporal features in the neural network. Our model achieves the fastest speed in experiments and comparable results. The frame-mAP result of JHMDB is shown in Table 4. Our method performs better than [36] but has a big gap with state-of-the-art. We obtain high accuracy on the training split

Table 4
Comparison (frame-mAP) to state-of-the-art on JHMDB. (The IoU threshold is set to 0.5).

Method	Detector	mAP@0.5
Weinzaepfel et al. [34]	R-CNN	45.8
Gkioxari et al. [36]	R-CNN	36.2
Peng et al. (w/o MR) [28]	Faster R-CNN	64.8
Hou et al. [35]	Tube Proposal	47.9
Ours	RFB Net	**37.4**

Table 5
Comparison (frame-mAP) to the state-of-the-arts on the AVA dataset. (The IoU threshold is set to 0.5). The last column indicates the frames per second (FPS). "*" means the method uses RGB and optical flow to detect action.

Method	mAP@0.5	FPS
Single frame* [15]	13.7	~ 5
I3D* [15]	15.6	~ 1
ARCN* [38]	17.4	~ 1
STEP [37]	18.6	21
Ours*	**15.2**	**37.8**

Table 6
Comparison (video-mAP) to the state-of-the-arts with different IoU thresholds on the UCF-101-24 dataset. The fifth column 0.5: 0.95 corresponds to the average video-mAP for the thresholds with a step of 0.05. The last column indicates the frames per second (FPS).

IoU threshold	0.2	0.5	0.75	0.5:0.95	FPS
Yu et al. [39]	26.5	–	–	–	–
Weinzaepfel et al. [34]	46.8	–	–	–	–
Saha et al. [17]	66.6	36.4	7.9	14.4	4
Peng(w/o MR) et al. [28]	71.8	35.9	1.6	8.8	–
Peng(w/ MR) et al. [28]	72.9	–	–	–	–
TPN (w/o LSTM) [9]	70.3	–	–	–	–
TPN (w/ LSTM) [9]	71.6	–	–	–	–
ROAD (w/ real-time OF) [3]	42.5	13.9	0.5	3.3	28
ROAD (w/ OF) [3]	73.5	46.3	15.0	20.4	7
Hou et al. [35]	47.1	–	–	–	–
Kalogeition et al. [18]	76.5	49.2	19.7	23.4	5.7
Yang et al. [37]	76.6	–	–	–	23
Ours	**74.8**	**46.6**	**16.7**	**21.9**	37.8

while getting a poor result on the testing split. The reason is that the small amount of videos in JHMDB leads to our model is unable to learn good feature representation. The results on the AVA dataset are shown in Table 5. Our method gets approximate result with other approaches in accuracy. While for efficiency, our model is much faster than the clip based [37] (21 fps) and the frame base [15] methods (5 fps). Fig. 5 shows our bounding boxes regression results.

4.6.2. Video-mAP

We report the video mAP results on the UCF-101-24 dataset in Table 6. We follow the protocol of [3] and present the results with different IoU thresholds 0.2, 0.5, 0.75. Specifically, 0.5: 0.95 denotes the average video-mAP corresponding to the threshold changes via a step size 0.05. At 0.2, our approach performs much better than [18,28,34,35,39] where with a large improvement by more than 25.2%, besides, it also outperforms the state-of-the-arts that rely on SSD [3]. Compared to [17,18,28,34,35,37,39], our exploited approach, which takes advantage of a supervised unified network with integrated flow subnet and multiscale feature blocks, exhibits better frame detection ability. For the previous works in Table 6, only [3,37] focused on the balance between speed and accuracy. Our method gets the fastest detection speed and comparable accuracy on the UCF-24 dataset.

Fig. 5. Examples of regressed bounding boxes (red) and the ground truth (green). We can find our model have a bad performance on the Scenarios that contain multiple people. But our model have good regression performance in single person scene.

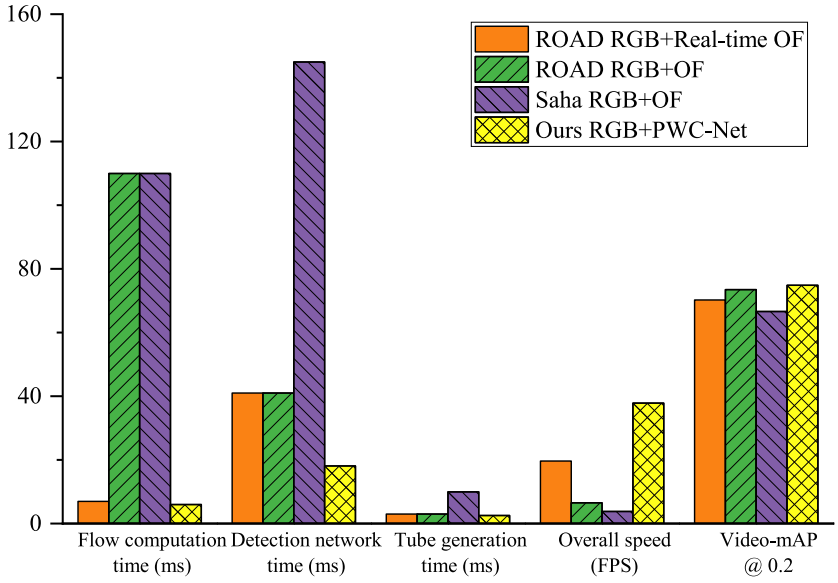

Fig. 6. Comparison of running speed. Real-time OF (optical flow) refers to the real-time optical flow method [40].

4.7. Running time analysis

We test the running speed of our method to evaluate whether it can localize human actions spatio-temporally in real-time with the usage of both RGB and optical flow. Previous action localization works normally use fast optical flow methods (*i.e.* real-time OF [20] and [40]) to generate pre-compute optical flow, in which the optical flow estimation and action localization are performed separately. However, this kind of method is far from being implemented in real-time. In contrast, we perform action

localization and motion estimation in a unified network. To be fair, we compare our approach with the state-of-the-art two-stream works [3,17] on a desktop with two Intel Xeon CPU @3.2GHz (totally 16 cores) and an NVIDIA GEFORCE GTX 1080Ti GPU. It should be noticed that the detection network cost time is twice larger than reported in [3], because Singh et al. utilizes two GPUs to compute RGB steam and optical flow (OF) stream respectively, we convert the cost time to one GPU scheme. We use the same tube generation algorithm as Singh et al. [3], which is efficient for both of our approach (2.5ms without boxes combining) and their model (3.0ms with boxes combining).

The results of the running speed of different methods are shown in Fig. 6. [3,17] cost much time on optical flow computation and object detection. In the real world applications, the computation time of the traditional optical flow methods [20,21] is unacceptable. Singh et al. run RGB stream SSD and flow stream SSD on two GPUs simultaneously, leading to it is low-efficient and is hard to combine detection boxes to perform video-level prediction. In contrast, our approach integrates optical flow estimation into the detection network and produces spatial and temporal features in the same network with a single GPU. The proposed method not only achieves a real-time speed of **37.8** FPS which is much faster than all the other methods but also obtains the highest accuracy. Our detection network costs 24.1ms to generate action detection boxes from both RGB and flow paths, which outperforms the currently most efficient and inefficient methods by respectively **1.9** times and **9.9** times speed improvement. Specially, compared to ROAD [3], which applies the classical Brox OF [20] and gets the highest efficiency to localize human action before us, our approach is **1.9** times faster (19.6 FPS vs 37.8 FPS) in implementation and with a **1.3%** video-mAP improvement. Compared to the low efficient methods [17] which use a two-stage action detector, our method is more than **9.9** times faster (3.8 FPS vs 37.8 FPS) and gets at least **8.2%** video-mAP promotion. Benefited from our unified network, optical flow is efficiently and accurately computed in the action localization framework jointly. Furthermore, the action bounding box is fast predicted due to spatial and temporal features can be computed and fused in the same network.

5. Conclusions

In this paper, we proposed a novel spatio-temporal action localization approach with integrated optical flow subnet to address the issues: 1) localizing human action instances with the usage of both RGB appearance and optical flow in real-time, 2) performing optical flow computation and action localization in one multi-task deep architecture jointly, and 3) demonstrating that these two tasks, which are performed interactively, are beneficial for each other. We conducted extensive experiments on a number of benchmark datasets (i.e. UCF101-24, JHMDB and AVA), and the results show that our approach gets better performance than existing approaches for online applications. We also provided experimental analysis to explain why our approach performs better than other approaches in speed and accuracy.

The structure of neural network plays an important role in speeding up the detection of human action in videos. Inspired by the recently novel CNNs and free-anchor object detection methods, in future work, we will explore other methods to improve the accuracy without slowing down the calculations. Specifically, we would like to replace the spatial and temporal backbone network with 3D CNN. Moreover, we plan to adopt a strategy similar to RPN, which can be optimized in a unified neural network, linking the actor-boxes to the tubes rather than using traditional object detection methods.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61702350. It was also supported by the National Key Research and Development Program of China (No. 2018YFB2100603) and the Wuhan University-Infinova project (No. 2019010019).

References

[1] P. Wang, W. Li, P. Ogunbona, J. Wan, S. Escalera, RGB-D-based human motion recognition with deep learning: a survey, Comput. Vis. Image Understand. 171 (2018) 118–139.
[2] L.R. Villegas, D. Colombet, P. Guiraud, D. Legendre, S. Cazin, A. Cockx, Image processing for the experimental investigation of dense dispersed flows: Application to bubbly flows, Int. J. Multiphase Flow 111 (2019) 16–30.
[3] G. Singh, S. Saha, M. Sapienza, P.H. Torr, F. Cuzzolin, Online real-time multiple spatiotemporal action localisation and prediction, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3637–3646.
[4] D. Zhang, M. Luo, F. He, Reconstructed similarity for faster GANs-based word translation to mitigate hubness, Neurocomputing 362 (2019) 83–93.
[5] D. Zhang, F. He, Z. Tu, L. Zou, Y. Chen, Pointwise geometric and semantic learning network on 3D point clouds, Integrat. Comput.-Aided Eng. 27 (1) (2020) 57–75.
[6] Z. Tu, N. Van Der Aa, C. Van Gemeren, R.C. Veltkamp, A combined post-filtering method to improve accuracy of variational optical flow estimation, Pattern Recog. 47 (5) (2014) 1926–1940.
[7] M. Zhai, X. Xiang, R. Zhang, N. Lv, A. El Saddik, Optical flow estimation using channel attention mechanism and dilated convolutional neural networks, Neurocomputing 368 (2019) 124–132.
[8] Z. Tu, W. Xie, D. Zhang, R. Poppe, R.C. Veltkamp, B. Li, J. Yuan, A survey of variational and CNN-based optical flow techniques, Signal Process. 72 (2019) 9–24.
[9] J. He, Z. Deng, M.S. Ibrahim, G. Mori, Generic tubelet proposals for action localization, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 343–351.
[10] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, Flownet: Learning optical flow with convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2758–2766.
[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, Flownet 2.0: Evolution of optical flow estimation with deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2462–2470.
[12] S. Liu, D. Huang, et al., Receptive field block net for accurate and fast object detection, in: European Conference on Computer Vision, Springer, 2018, pp. 385–400.
[13] R.-C. Chen, et al., Automatic license plate recognition via sliding-window darknet-YOLO deep learning, Image Vis. Comput. 87 (2019) 47–56.
[14] Z. Tu, W. Xie, Q. Qin, R. Poppe, R.C. Veltkamp, B. Li, J. Yuan, Multi-stream CNN: Learning representations based on human-related regions for action recognition, Pattern Recognit. 79 (2018) 32–43.
[15] C. Gu, C. Sun, D.A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al., Ava: a video dataset of spatio-temporally localized atomic visual actions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6047–6056.
[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[17] S. Saha, G. Singh, F. Cuzzolin, Amtnet: action-micro-tube regression by end–to-end trainable deep architecture, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4414–4423.
[18] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, C. Schmid, Action tubelet detector for spatio-temporal action localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4405–4413.
[19] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. (6) (2017) 1137–1149.
[20] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: European conference on computer vision, Springer, 2004, pp. 25–36.
[21] D. Sun, S. Roth, M.J. Black, A quantitative analysis of current practices in optical flow estimation and the principles behind them, Int. J. Comput. Vis. 106 (2) (2014) 115–137.
[22] P. Zhang, W. Liu, Y. Lei, H. Lu, Hyperfusion-net: Hyper-densely reflective feature fusion for salient object detection, Pattern Recognit. 93 (2019) 521–533.

[23] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.
[24] K. Ahmad, N. Conci, How deep features have improved event recognition in multimedia: a survey, ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) 15 (2) (2019) 1–27.
[25] D. Sun, X. Yang, M.-Y. Liu, J. Kautz, Models matter, so does training: An empirical study of CNNs for optical flow estimation, IEEE Trans. Pattern Anal. Mach. Intell. (2019), doi:10.1109/TPAMI.2019.2894353. 1–1
[26] S. Megrhi, M. Jmal, W. Souidene, A. Beghdadi, Spatio-temporal action localization and detection for human action recognition in big dataset, J. visual Commun. Image Represent. 41 (2016) 375–390.
[27] H. Kuehne, A. Richard, J. Gall, Weakly supervised learning of actions from transcripts, Comput. Vis. Image Understand. 163 (2017) 78–89.
[28] X. Peng, C. Schmid, Multi-region two-stream R-CNN for action detection, in: European conference on computer vision, Springer, 2016, pp. 744–759.
[29] Y. Li, L. Jia, Z. Wang, Y. Qian, H. Qiao, Un-supervised and semi-supervised hand segmentation in egocentric images with noisy label learning, Neurocomputing 334 (2019) 11–24.
[30] I. Úbeda, J.M. Saavedra, S. Nicolas, C. Petitjean, L. Heutte, Improving pattern spotting in historical documents using feature pyramid networks, Pattern Recognit. Lett. 131 (2020) 398–404.
[31] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2017) 1510–1517.
[32] W. Sultani, M. Shah, Automatic action annotation in weakly labeled videos, Comput. Vis. Image Understand. 161 (2017) 77–86.
[33] M. Everingham, S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (1) (2015) 98–136.
[34] P. Weinzaepfel, Z. Harchaoui, C. Schmid, Learning to track for spatio-temporal action localization, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 3164–3172.
[35] R. Hou, C. Chen, M. Shah, Tube convolutional neural network (T-CNN) for action detection in videos, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5822–5831.
[36] G. Gkioxari, J. Malik, Finding action tubes, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 759–768.
[37] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L.S. Davis, J. Kautz, Step: spatio-temporal progressive learning for video action detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 264–272.
[38] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, C. Schmid, Actor-centric relation network, in: European conference on computer vision, Springer, 2018, pp. 318–334.
[39] G. Yu, J. Yuan, Fast action proposals for human action detection and search, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1302–1311.
[40] N. Afrashteh, S. Inayat, M. Mohsenvand, M.H. Mohajerani, Optical-flow analysis toolbox for characterization of spatiotemporal dynamics in mesoscale optical imaging of brain activity, Neuroimage 153 (2017) 58–74.

Dejun Zhang received the bachelor in communication engineering at the School of Information Science and Technology, Southwest Jiaotong University, China, 2006. In 2011, he received the Master degree in electronic engineering at the School of Manufacturing Science and Engineering, Southwest University of Science and Technology, China. In 2015, he received the Ph.D. degree in Computer Science from Wuhan Univeristy, China. He is currently a lecturer with the faculty of School of Information Engineering, China University of Geosciences, Wuhan, China. His research areas include machine learning, bioinformatics and computer graphics. His research interests include digital geometric processing, computer graphic, action recognition and localization.

Linchao He is a Bachelor of the College of Information Engineering, Sichuan Agricultural University. His research areas include machine learning, natural language processing and computer vision. His research interests include neural network optimization, action recognition and detection.

Zhigang Tu started his Master Degree in image processing at the School of Electronic Information, Wuhan University, China, 2008. 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. Then from 2016 to 2018, he was a research fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently a professor at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University. His research interests include computer vision, image processing, video analytics, and machine learning. Specially for motion estimation (Optical Flow), object segmentation, object tracking, action recognition and localization, anomaly detection, etc.

Boxiong Yang received the bachelor in computer software at the School of Computer Science and Technology, Huazhong Normal University, China, 1997. In 2000, he received the Master degree in Geodesy at Institute of Seismology, China Seismological Bureau. In 2005, he received the Ph.D. degree in Institute of Geophysics, China Seismological Bureau. He is currently an associate professor at School of Information and Intelligence Engineering, University of Sanya, Sanya, China. His research areas include Artificial Intelligence and computer vision. His research interests include image recognition, video classification, action recognition and localization.