Neural Computing and Applications (2021) 33:5167–5181
https://doi.org/10.1007/s00521-020-05313-8

ORIGINAL ARTICLE

Multi-cue based 3D residual network for action recognition

Ming Zong[1,2] · Ruili Wang[1,2] · Zhe Chen[3] · Maoli Wang[4] · Xun Wang[1,2] · Johan Potgieter[5]

Received: 19 March 2020 / Accepted: 19 August 2020 / Published online: 2 September 2020

Abstract
Convolutional neural network (CNN) is a natural structure for video modelling that has been successfully applied in the field of action recognition. The existing 3D CNN-based action recognition methods mainly perform 3D convolutions on individual cues (e.g. appearance and motion cues) and rely on the design of subsequent networks to fuse these cues together. In this paper, we propose a novel multi-cue 3D convolutional neural network (M3D), which integrates three individual cues (i.e. an appearance cue, a direct motion cue, and a salient motion cue) directly. Different from the existing methods, the proposed M3D model directly performs 3D convolutions on multiple cues instead of a single cue. Compared with the previous methods, this model can obtain more discriminative and robust features by integrating three different cues as a whole. Further, we propose a novel residual multi-cue 3D convolution model (R-M3D) to improve the representation ability to obtain representative video features. Experimental results verify the effectiveness of proposed M3D model, and the proposed R-M3D model (pre-trained on the Kinetics dataset) achieves competitive performance compared with the state-of-the-art models on UCF101 and HMDB51 datasets.

Keywords Action recognition · Multi-cue · 3D convolution · Salient motion cue · Residual

1 Introduction

Human action recognition aims to automatically identify specified actions in a video [29, 30]. It has many applications such as intelligent video surveillance, human-computer interaction, human behaviour analysis, and smart hospital care [32, 34, 37, 40, 41, 44]. Different from images that only contain an appearance cue, videos contain not only the appearance cue extracted from still video frames but also a motion cue extracted from stacked video frames. Therefore, the motion cue plays an important role in action recognition.

Compared with traditional shallow hand-crafted models [53, 54], convolutional neural networks (CNNs) [47, 48, 59, 61] have shown a superior ability to capture appearance information in many visual-related tasks such as image classification [27, 58], object detection [13], and image segmentation [31].

To take advantage of CNNs, many 2D CNN-based methods [4, 9, 25, 45, 51, 57] were developed for action recognition. 2D CNN-based action recognition models can be roughly divided into two categories: (i) frame-based aggregation models [4, 9, 25, 57], and (ii) two-stream CNN-based models [45]. Frame-based aggregation models use CNNs to extract features from each frame and then aggregate frame-level information to obtain video-level information by using different aggregating strategies such as recurrent neural networks (RNNs). However, such models ignore the temporal structure and mainly rely on subsequent aggregation strategies for obtaining a motion cue. Two-stream CNN-based models consist of a spatial CNN stream and a temporal CNN stream for capturing the appearance cue and motion cue, respectively. Optical flow-based methods usually are adopted for extracting the

✉ Maoli Wang
wangml@qfnu.edu.cn

1 School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China

2 School of Natural and Computational Sciences, Massey University, Auckland, New Zealand

3 College of Computer and Information, Hohai University, Nanjing, China

4 School of Information Science and Engineering, Qufu Normal University, Rizhao 276800, China

5 School of Food and Advanced Technology, Massey University, Auckland, New Zealand

motion features, which can effectively provide the velocity information (including the speed and direction) of each pixel. However, such models lack effective information interaction over time between the appearance cue and motion cue.

Recently, a variety of 3D CNN-based action recognition models have been developed for modelling spatiotemporal features [23, 49], which has shown more promising results than the previous CNN-based models for action recognition on a sufficiently large video dataset such as Kinetics dataset [4, 17]. According to the input cue, we categorize the existing 3D CNN-based action recognition models into three classes: (i) single-cue 3D CNN model [25], (ii) two-cue 3D CNN model [45] and (iii) three-cue 3D CNN model [23].

- *Single-cue 3D CNN model:* The input of a single-cue 3D CNN model is stacked video frames that only provide an appearance cue at the input level. The motion information is indirectly obtained by 3D convolutions through the inferences between stacked video frames. Thus, such models lack the ability to provide direct motion cues at the input level [45].
- *Two-cue 3D CNN model* Two-cue 3D CNN models [4, 52] perform 3D convolutions on video frames and optical flow frames separately, and then the obtained appearance and motion cues are integrated by using various fusion strategies. However, since the 3D convolutions are separately operated on individual cues respectively and the fusion relies on specific designed networks, this model lacks the overall integration of the appearance information and motion information of videos.
- *Three-cue 3D CNN model* Three-cue 3D CNN models [23] generate different cues (such as gray, gradient, and optical flow) from stacked video frames, and performed 3D convolutions on these cues separately. Then a full connection layer is applied to combine all cues. Late fusion is also adopted for aggregating different cues for action recognition. Although optical flow [3, 45] is capable of capturing motion information (*i.e.* speed and direction information) directly, it mainly focuses on instantaneous motion velocity information that is sensitive to background motion noises such as slight leaf jittering and water rippling [20, 33].

To overcome the above challenges, we propose a novel multi-cue 3D convolutional neural network (M3D). The proposed M3D model integrates an appearance cue, a direct motion cue and a salient motion cue simultaneously as the input. The appearance cue can extract features from still video frames, the motion cue can extract features from stacked video frames, and the salient motion cue can effectively suppress background motion noises and highlight salient motion. Different from the previous action recognition methods (single-, two-, and three-cue CNN models), our proposed M3D model directly performs 3D convolutions on a multi-cue input instead of performing 3D convolutions on different single-cue inputs separately. Further, we propose a novel residual multi-cue 3D convolution model (R-M3D) to improve the representation ability to obtain representative video features.

The key contributions of this paper can be summarized as follows:

- This paper proposes a novel M3D model, which first directly performs 3D convolutions on a multi-cue input instead of a single-cue input.
- This paper develops a novel video frame representation for performing 3D convolutions on a multi-cue input.
- The proposed M3D model adds the salient motion cue for action recognition, which can suppress background motion noises and highlight salient motion.
- To further improve the representation ability, we propose the deeper R-M3D model based on 3D ResNet, which can significantly improve the performance of action recognition.

The rest of the paper is presented as follows: background and related work are introduced in Sect. 2. Then, two novel multi-cue 3D convolutional neural network (M3D and R-M3D) are developed in Sect. 3. Experimental results and analysis will be discussed in Sect. 4. Lastly, conclusion and future work will be summarized in Sect. 5.

2 Background and related work

A video contains both time and space information. With the development of deep learning techniques, deep neural network-based action recognition methods, especially CNN-based action recognition methods, have obtained better performances than the conventional shallow bag-of-visual-word- based models [23]. 2D CNNs and 3D CNNs are two types of neural networks commonly used in CNN-based action recognition methods.

Section 2.1 first introduces some 2D CNN-based action recognition works. Then, we present the general 3D CNNs architecture and some 3D CNN-based action recognition works in Sect. 2.2. Finally, we present the principle of 3D convolutions performed on a single-cue input for action recognition in Sect. 2.3.

2.1 2D CNN for action recognition

2D CNN-based action recognition methods usually extract features from single video frame using 2D CNNs, and then aggregate temporal information across video frames using

different fusion strategies. In general, these methods average the output of the utilized 2D CNNs on each frame and utilise the average frame-level result to represent video-level results [25]. However, these methods cannot make full use of video information. They only extract appearance information from videos while ignoring motion information between video frames.

To capture temporal motion information between video frames, two-stream CNN-based models were developed for action recognition. They usually consist of two streams. The spatial stream is responsible for capturing the appearance cue and the temporal stream is responsible capturing the motion cue. Then various fusion strategies are adopted to aggregate the output of these two streams [10–12, 45]. In addition, some methods utilize recurrent neural networks (RNN) [4, 9, 57], especially long short-term memory (LSTM) networks, were utilized to capture temporal information between stacked video frames. A pipeline of RNN-based action recognition methods is that (i) 2D CNNs are used for extracting features from each video frame, and (ii) LSTM networks are used for encoding states and learning temporal relations between stacked video frames.

2.2 3D CNN for action recognition

In contrast to 2D CNN for action recognition, some action recognition methods utilize 3D CNNs to model spatiotemporal features from videos. A common architecture of 3D CNNs for action recognition can be illustrated in Fig. 1, which consists of five convolutional layers, two fully connected layers, and a softmax layer. Different from the input of 2D CNNs that is a single frame, the input of 3D CNNs is stacked video frames. Different from 2D CNNs, which only slide a convolution filter kernel along the spatial direction, the 3D convolution filter kernel slides along with both spatial and temporal directions simultaneously. Thus, 3D CNNs can naturally capture spatiotemporal information, while 2D CNNs need to aggregate multiple frame-level information over long time periods to capture the temporal information of videos.

It has been proven that 3D CNN-based models have shown competitive performance than 2D CNN-based models [17]. Early work of 3D CNN-based action recognition models were proposed by Baccouche et al. [2], which extracted 3D CNN-based features for each frame and input them sequentially into a LSTM network for classification. Ji et al. [23] proposed to separately perform 3D convolutions on multiple channels. Then, a late fusion was used for all these channels. These 3D CNNs are not obviously deep, which typically contain 3 convolutional layers at most.

Recently, a variety of deeper 3D CNN-based models were developed. Karpathy et al. [25] proposed a slow fusion strategy to aggregate frame-level information across temporal domain. Tran et al. [49] proposed to use a small $3 \times 3 \times 3$ convolution kernel in all convolutional layers and got better performance than other large sizes of convolution kernels. Similar to [5], 3D convolutions were performed separately on a spatial stream and a temporal stream in [4], the output results of two streams were aggregated later. Hara et al. [16] proposed to use a very deep 3D residual network instead of shallow 3D convolutional networks. Further, deep 3D CNNs architectures on sufficiently large datasets can get better performances than those complex 2D CNNs architectures [17]. Arunnehru et al [1] extracted motion information from the video and regarded them as 3D motion cuboid and then applied a 3D convolutional neural network for action recognition. Zhou et al [62] proposed a novel Mixed Convolutional Tube to combine 2D CNNs with the 3D convolution together for action recognition, which can generate more discriminative feature maps. Qiu et al [43] considered using a $1 \times 3 \times 3$ spatial convolution kernel plus a temporal $3 \times 1 \times 1$ convolution kernel to simulate the previous $3 \times 3 \times 3$ convolution kernel, which is helpful for reducing the computational cost and training deeper neural networks. Based on 3D ResNets, Tran et al [50] decomposed 3D

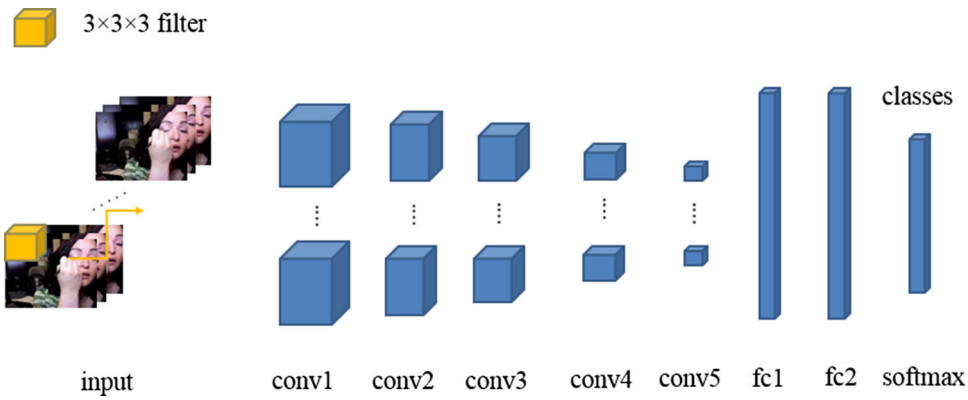

Fig. 1 The architecture of 3D CNN for action recognition, which consists of five convolutional layers, two fully connected layers, and a softmax layer. The kernel size is $3 \times 3 \times 3$

convolution kernel into spatial convolution and temporal convolution and designed a novel spatiotemporal convolutional block R(2+1)D, which can achieve superior performance for action recognition compared with the state of the art.

Although current 3D convolution models can achieve competitive performance, most of them only directly perform 3D convolutions on a single-cue input, e.g. on stacked video frames, and optical flow frames, separately. To obtain richer information in the input level such as motion velocity and motion saliency, we propose to directly perform 3D convolutions on a multi-cue input in Sect. 3.

2.3 3D convolutions performed on a single-cue input

Current 3D convolutions usually are performed on a single-cue input, i.e. stacked video frames or stacked optical flow frames. Specifically, taking stacked video frames as an example, we denote n stacked video frames as $\{vf_1, vf_2, \ldots, vf_n\}$ An image in the RGB format usually consists of three different colour channels: the red channel, green channel, and blue channel. Thus each video frame vf can be represented as $\{vf\_r, vf\_g, vf\_b\}$, where vf_r represents a red channel; vf_g represents a green channel, and vf_b represents a blue channel. Correspondingly, the stacked video frames can be represented as $\{vf_1\_r, vf_1\_g, vf_1\_b, \ldots, vf_n\_r, vf_n\_g, vf_n\_b\}$, where n denotes the number of input video frames. The number of total channels of the input stacked video frames is $n \times 3$. The computation process of 3D convolutions performed on a single-cue input (*i.e.* stacked video frames) is illustrated in Fig. 2.

In the temporal dimension, as a $3 \times 3 \times 3$ filter kernel is applied, total 9 colour channels are involved for each computation process of 3D convolutions performed on a single-cue input. Three stacked video frames are involved in each computation, and each video frame contains three different colour channels. Since 3D convolutions are only applied to the single-cue input, *i.e.* stacked video frames, it lacks the ability to provide direct motion information in the input level. For two-cue 3D CNN models, they still separately perform 3D convolutions on individual single-cue input such as stacked video frames or stacked optical flow frames, and then a late fusion is adopted. However, late fusion is lack of cues evolution over time.

3 Our models: M3D & R-M3D

In this section, we will first present a motion saliency detection method to suppress background motion noises and provide salient motion cue, which will be used in our proposed M3D model. Then, we present the novel triple video representation in Sect. 3.2. Based on novel triple video representation, 3D convolutions performed on a multi-cue input are proposed in Sect. 3.3. Finally, a novel M3D model is presented in Sect. 3.4 and a novel deep R-M3D model is presented in Sect. 3.5.

3.1 Motion saliency detection

Saliency detection aims to detect the salient object from the whole image. For example, Ji et al [24] proposed to use an attention CNN layer to capture the context information between different feature maps to improve the quality of obtained saliency maps. In contrast to traditional saliency detection methods that identify salient objects on a still image, motion saliency detection methods [5, 6, 8, 56] focus on identifying salient motion information (named motion saliency for short [6]) from a video. However, conventional motion saliency detection methods have expensive time cost and are not suitable for large video datasets. Compared with conventional motion saliency detection methods [7, 18, 22], spectrum-based motion saliency detection methods have a good balance between performance and computation cost [6].

The spectral theory has been widely used in the saliency detection field [6, 8, 15, 21]. The first spectrum-based saliency detection work was proposed by Hou et al. [21],

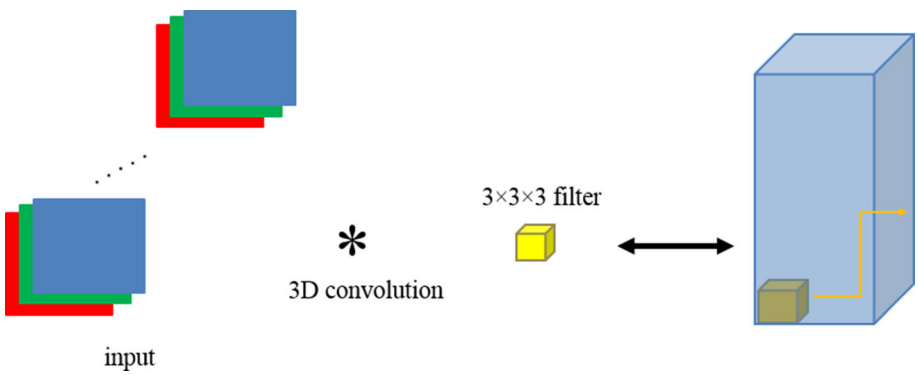

Fig. 2 The computation process of 3D convolutions performed on a single-cue input, i.e. stacked video frames. The red rectangle represents the red colour channel, the green rectangle represents the green colour channel, and the blue rectangle represents the blue colour channel

which used amplitude spectral residual of an image to represent the novelty part. Guo et al. [15] proposed to use the phase spectrum of an image to represent the novelty part instead of the amplitude spectrum. According to Spectral Residual (SR) theory [21], the image information contains the novelty part and the redundant information. Therefore, an image can be represented as follows [21]:

$$\gamma(Image) = \gamma(Innovation) + \gamma(PriorKnowledge) \quad (1)$$

where $\gamma(Image)$ represents the image information; $\gamma(PriorKnowledge)$ represents the redundant information, which denotes the statistical invariant properties in the image; $\gamma(Innovation)$ represents the novelty part, which denotes the statistical variant properties in the image [21]. In the field of motion saliency detection, $\gamma(Innovation)$ corresponds to the foreground objects, and $\gamma(PriorKnowledge)$ corresponds to the background [6, 15].

In this paper, a phase spectrum-based motion saliency detection method [6] is introduced for capturing the salient motion information and suppressing the background motion noises. It distinguishes the salient motion and background motion noises by identifying phase spectrum variations of each pixel through a temporal Fourier transform. The key procedure of this method can be summarized as follows:

Procedure:	Phase spectrum-based motion saliency detection.
Step1	Establishing a temporal sequence $S_{x,y}(t)$ for each pixel in the same position (x, y) through stacked video frames, here the size of a video frame is $M \times N$, $x = 1, 2, \ldots, M$, $y = 1, 2, \ldots, N$, and $t = 1, 2, \ldots, T$. Thus $M \times N$ temporal sequences are established.
Step2	Calculating the Fourier transform for each temporal sequence $S_{x,y}(t) : f_{x,y}(t) = F(S_{x,y}(t))$, here F denotes the Fourier transform.
Step3	Calculating phase spectrum $p_{x,y}(t)$ for each temporal sequence $S_{x,y}(t) : p_{x,y}(t) = angle(f_{x,y}(t))$, here *angle* denotes a function of obtaining phase values of a temporal sequence.
Step4	Calculating inverse Fourier transform $f^{-1}_{x,y}(t)$ for each temporal sequence $S_{x,y}(t) : f^{-1}_{x,y}(t) = g(t) * F^{-1}(p_{x,y}(t))$, here F^{-1} denotes inverse Fourier transform and $g(t)$ denotes a one-dimensional Gaussian filter for smoothing noises.

Through the above steps, the salient motion can be identified by obvious phase spectrum variations, and noises can be identified by slight phase spectrum variations. Therefore, background and background motion noises are easily suppressed because the corresponding values of the phase spectrum are much smaller than the values produced by salient motions. Figure 3 illustrates a video frame extracted from a makeup video clip from the UCF101 dataset [46], and its corresponding optical flow and motion saliency results.

3.2 Novel triple video representation

A traditional video representation consists of stacked video frames, i.e. stacked video frames are used to describe video information. However, this video representation manner only can provide the appearance cue as input for current single-cue 3D CNN models in action recognition, which lacks the ability to provide a direct motion cue in the input level.

For two-cue 3D CNN models, they usually adopt video frames and optical flow frames as input to provide the appearance cue and the direct motion cue, i.e. they use video frame and optical flow to describe video information. However, since they separately use video frames and optical flow frames as input to operate 3D convolutions, this manner relies on specific designed networks to fuse the appearance cue and motion cue instead of directly operating 3D convolutions on a multi-cue input. Besides, optical flow-based motion detection methods are sensitive to background motion noises.

To overcome the aforementioned challenges in current 3D CNN models, we develop a novel triple video representation composed of the original video frame, the corresponding optical flow and motion saliency, which is illustrated in Fig. 4. The proposed triple video representation integrates an appearance cue, a direct motion cue and a salient motion cue simultaneously as the input. The appearance cue can extract features from still video frames, the motion cue can extract features from stacked video frames, and the salient motion cue can effectively suppress background motion noises and highlight salient motion.

The triple video representation consists of video frame, optical flow and motion saliency. The stacked video frames $\{vf_1, vf_2, \ldots, vf_n\}$ provide the appearance cue. The optical flow provides direct motion information instead of the inferences between stacked video frames. A popular optical flow-based motion detection method [3] is used to extract optical flow $\{opf_1\_x, opf_2\_x, \ldots, opf_n\_x\}$ and $\{opf_1\_y, opf_2\_y, \ldots, opf_n\_y\}$ from the stacked video frames, here opf_i_x denotes the horizontal optical flow of the *ith* video frame and opf_i_y denotes the vertical optical flow of the *ith* video frame. The motion saliency captures the salient motion information and suppresses background motion noises. The phase spectrum-based motion saliency detection method [6] is used to obtain motion saliency information $\{ms_1, ms_2, \ldots, ms_n\}$ from the n stacked video frames, here ms_i denotes the motion saliency of the *ith* video frame.

Fig. 3 The left picture denotes a video frame; the middle picture denotes the corresponding optical flow result, and the right picture denotes the corresponding motion saliency result

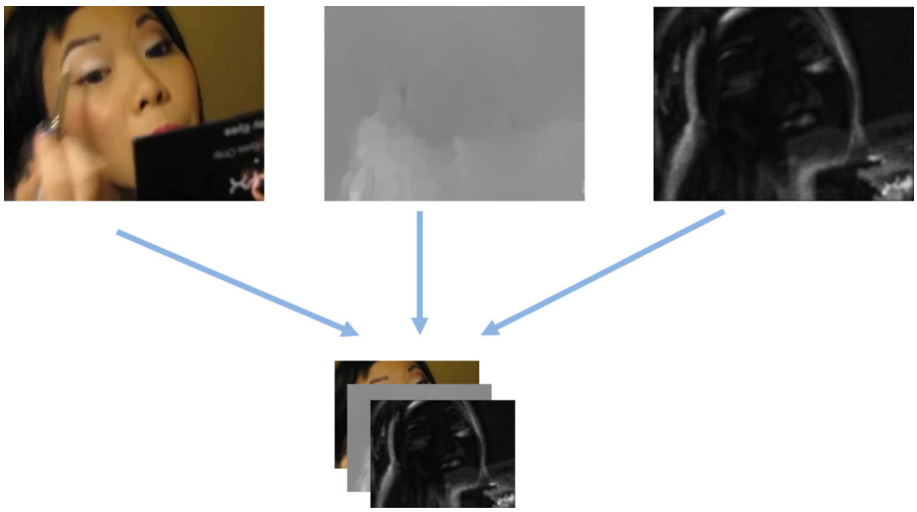

Fig. 4 The triple video representation consists of video frame, optical flow and motion saliency

Based on the above fundamentals, the triple video frame can be represented as $tvf : \{vf, opf, ms\}$, i.e. $\{vf_r, vf_g, vf_b, opf_x, opf_y, ms\}$. Compared with the previous video frame *vf* which only provides the appearance information as an input, the novel triple video frame *tvf* can provide richer input information, including appearance information, direct motion information and salient motion information.

3.3 3D convolutions on a multi-cue input

Based on the novel triple video representation, we propose to perform 3D convolutions on the stacked triple video frames, i.e. perform 3D convolutions on a multi-cue input instead of a single-cue input. Three different cues are adopted as an input, which includes the appearance cue, direct motion cue and salient motion cue. The computation process of the proposed novel 3D convolutions on a multi-cue input can be illustrated in Fig. 5.

From the perspective of the temporal dimension, as a $3 \times 3 \times 3$ filter kernel is applied, total 18 colour channels are used for each computation of the proposed 3D convolution on a multi-cue input. Three stacked triple video frames are involved for each computation, and each triple video frame contains 6 channels: three colour channels, a horizontal optical flow channel, a vertical optical flow channel and a motion saliency channel. An obvious difference between the single-cue input and the multi-cue input is the number of input channels. The single-cue input of stacked video frames usually provides 3 input channels while the proposed multi-cue input usually provides 6 input channels. The added channels focus on providing motion information, including motion velocity and motion saliency.

3.4 M3D model

To evaluate the effectiveness of our proposed 3D convolutions on a multi-cue input, we design our multi-cue 3D convolution based on C3D model (M3D for short) [49]. The C3D model is a benchmark architecture of 3D convolution neural networks, which contains 8 convolutional layers, 5 max-pooling layers, 2 fully connected layers and 1 softmax layer. The architecture can be denoted as (conv1, pool1, conv2, pool2, conv3, conv4, pool3, conv5, conv6, pool4, conv7, conv8, pool5, fc1, fc2, and softmax). The main difference between the C3D model and our proposed M3D model is in the input part. The C3D model performs 3D convolutions on a single-cue input, i.e. each video clip as a sample. However, our proposed M3D model performs 3D convolutions on a multi-cue input using the proposed triple video frame input representation, which stacks video frames (including RGB three colour channels), optical flow

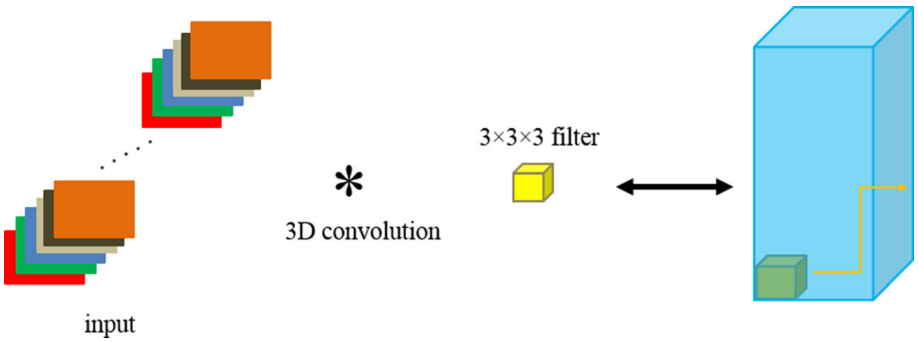

Fig. 5 An illustration of the 6 involved channels for each computation of the proposed novel 3D convolutions on a multi-cue input along the temporal direction. The 6 channel are shown in different colours. The red rectangle, green rectangle and blue rectangle represent three different colour channels. The shallow grey rectangle and dark grey rectangle represent the corresponding horizontal optical flow channel and the vertical optical flow channel. The orange rectangle represents the corresponding motion saliency channel

frames (including horizontal and vertical two directions channels) and motion saliency frames (including one channel) together as one input. The architecture of our proposed M3D model can be illustrated in Fig. 6.

The procedure of the M3D model can be summarized in Algorithm 1 as follows:

Algorithm 1:	M3D model
Step1:	Extracting optical flow $\{opf_1\_x, opf_2\_x, ..., opf_n\_x\}$ and $\{opf_1\_y, opf_2\_y, ..., opf_n\_y\}$ from the stacked video frames $\{vf_1, vf_2, ..., vf_n\}$.
Step2:	Extracting motion saliency $\{ms_1, ms_2, ..., ms_n\}$ from the stacked video frames $\{vf_1, vf_2, ..., vf_n\}$.
Step3:	Expanding each video frame as a triple video frame representation composed of the original video frame, the corresponding optical flow and motion saliency. The triple video frame can be represented as tvf : $\{vf_r, vf_g, vf_b, opf_x, opf_y, ms\}$.
Step4:	Obtaining the corresponding stacked triple video frames$\{tvf_1, tvf_2, ..., tvf_n\}$ from the stacked video frames $\{vf_1, vf_2, ..., vf_n\}$.
Step5:	Performing 3D convolutions on the stacked triple video frames $\{tvf_1, tvf_2, ..., tvf_n\}$ and training our proposed M3D network for classification.

3.5 Deep R-M3D model

To improve the representation ability to obtain more representative video features. We further explore to apply our proposed multi-cue 3D convolutions on deep network architectures. 3D Residual Network (3D ResNet for short) [16, 19] is an excellent deep network, which can effectively alleviate the degradation problem [19]. The degradation problem indicates that as the network depth increasing, the training accuracy will get saturated but then degrade rapidly. Standard deep neural networks usually use multiple stacked layers to approximate the desired underlying mapping and transit information layer by layer. Formally, a building block of stacked layers in standard deep neural networks can be defined as follows:

$$H(\mathbf{x}) \approx F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\}) \tag{2}$$

where $\mathbf{x}$ denotes the input, $H(\mathbf{x})$ denotes the desired underlying mapping function, $F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\}$ denotes the actual output function of multiple stacked nonlinear layers

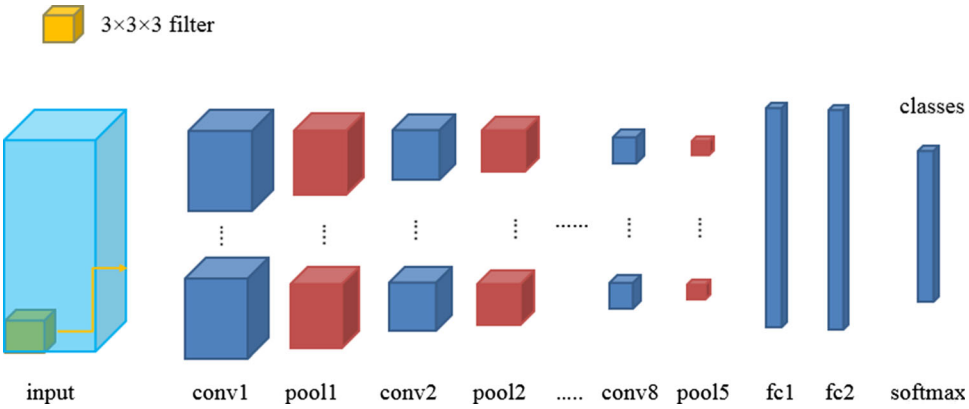

Fig. 6 The architecture of the proposed M3D model, which can be denoted as (conv1, pool1, conv2, pool2, conv3, conv4, pool3, conv5, conv6, pool4, conv7, conv8, pool5, fc1, fc2, softmax)

which can approximate the desired underlying mapping function $H(\mathbf{x})$, $\mathbf{W}$ denotes the weights and $\mathbf{b}$ denotes the biases. For example, a building block of two stacked layers is illustrated in Fig. 7. We can find that $F = W_2 f(W_1\mathbf{x} + b_1) + b_2$ where $f(\cdot)$ denotes the nonlinear activation function ReLU [19].

In contrary to standard deep neural networks, ResNets consider using multiple stacked layers to approximate a residual mapping by directly pass gradient flows from front layers to back layers, which can effectively ease the degradation problem. Formally, a residual building block can be defined as follows:

$$H(\mathbf{x}) \approx F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\}) + \mathbf{x} \tag{3}$$

where $H(\mathbf{x}) - \mathbf{x}$ denotes the residual mapping function and $F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\}) + \mathbf{x}$ is regarded as injecting a shortcut connection from the input to the output. For example, a residual building block of ResNets can be illustrated in Fig. 8.

Based on 3D ResNet, we design a deep residual multi-cue 3D convolution model (R-M3D for short). Compared with the M3D model consisting of 8 layers, the number of layers of R-M3D can reach to {18, 34, 50, 101 and 152}. Similar to the M3D model based on C3D model, the main difference between R-M3D and 3D ResNet depends on whether 3D convolutions operating on a multi-cue input or a single-cue input. In addition, in contrast to 3D ResNet directly using a fully connected layer as the output layer (i.e. the softmax layer), we add a new fully connected layer between the last convolutional layer and the output layer in our proposed R-M3D, which can synthesize all feature maps of the last convolutional layer together to improve the performance of action recognition, more details can be found in Sect. 4.2.1. The procedure of R-M3D is similar to the procedure of M3D model as illustrated in Algorithm1.

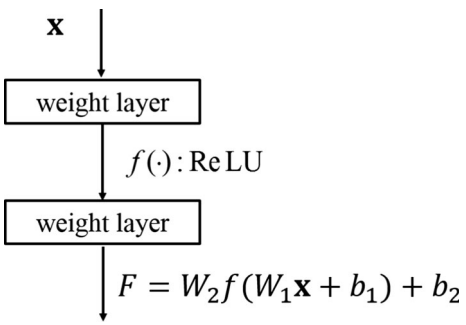

Fig. 7 The building block of standard deep neural networks

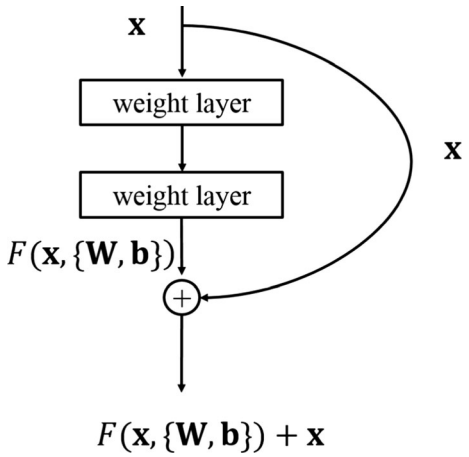

Fig. 8 The residual building block of ResNets

4 Experimental analysis

In this section, we first introduce the UCF101 and HMDB51 datasets in Sect. 4.1. Then, discuss the implementation for network architecture, training process and recognition process in Sect. 4.2. After that, we mainly analysis the experimental results of the single-cue model, two-cue models and the proposed M3D model in Sect. 4.3. Finally, the experimental results of the proposed deep R-M3D model will be presented and compared with other state-of-the-art models in Sect. 4.4.

4.1 Datasets

We mainly evaluate our model and other models on UCF101 [46] and HMDB51 [28] datasets.

- UCF101 dataset is a temporal trimmed action dataset, which contains 13320 videos and has 101 action categories (average about 130 videos for each action category). Three train/test splits are provided for distinguishing training dataset and test dataset. There are about 9500 videos for training and about 3800 videos for testing according to each UCF101 split.
- HMDB51 dataset contains 6766 videos and 51 action categories, each action category contains at least 101 videos. These action categories can be grouped into 5 different types: general facial actions such as smile, facial actions with object manipulation such as smoke, general body movements such as climb, body movements with object interaction such as golf, body movements for human interaction such as hug. Similar to UCF101 dataset, this dataset also provides three train/test splits for distinguishing training dataset and test dataset (about 70% training dataset and 30% test dataset).

4.2 Implementation

The experimental environment is Ubuntu 16.04 and Python 3.5. We implement our proposed M3D model and R-M3D model based on the deep learning framework Pytorch 0.3.1. To train our proposed M3D model and R-M3D model on UCF101 and HMDB51 datasets, stochastic gradient descent (SGD) is adopted. We train our model with a batch size of 32 on 4 GPUs (Nvidia GTX 1080Ti). The learning rate is set to 0.1 and it will be divided by 10 if the validation loss saturates. The weight decay is set to 0.001. The network architecture of the proposed M3D model can be found in Sect. 3.4. We present the network architecture of the proposed R-M3D model in Sect. 4.2.1. Then the training process and recognition process in detail are presented in Sects. 4.2.2 and 4.2.3, respectively.

4.2.1 Network architecture of R-M3D

To explore a deep network, we choose a deep 3D ResNet [17] as the backbone network instead of C3D network. According to [17], 3D ResNet with 34 layers is enough and suitable for UCF101 and HMDB51 datasets. In our implementation, we adopt 3D ResNet with 34 layers and modify 3D ResNet into R-M3D, the main differences between them contains two aspects: (i) The input channels of R-M3D are 6 (a multi-cue triple video input) while the input channels of 3D ResNet are 3 (a single-cue RGB input), thus the proposed R-M3D perform 3D convolutions on a multi-cue input; (ii) A new fully connected layer of 1024 neurons is added between the last convolutional layer and the output layer (*i.e.* the softmax layer) in our proposed R-M3D, which can synthesize all feature maps of the last convolutional layer together to improve the performance of action recognition. The number of neurons in the softmax layer is 101 or 51, which is corresponding to the number of classes on UCF101 or HMDB51 datasets, respectively. Thus, the proposed R-M3D contains 35 layers, which can be denoted as (conv1, maxpool, conv2_3, conv3_4, conv4_6, conv5_3, averagepool, fc and softmax), here x in conv2, 3, 4, 5_x denotes the multiple of the corresponding residual building block. Each frame is resized into 112 $\times$ 112 and the input size of each sample clip is 16 $\times$ 6 $\times$ 112 $\times$ 112, where 16 denotes each sample clip contains 16 frames and 6 denotes the number of channels (more detail can be found in Sect. 4.2.2). The detail of R-M3D network architecture is illustrated in Table 1.

4.2.2 Training process

To perform data augmentation, we first randomly select a video clip of 16 consecutive frames as a training sample from a raw video. Then similar to [55], the training sample is randomly cropped from 5 positions: top left corner, top right corner, bottom left corner, bottom right corner and centre. We also randomly horizontally flip the training sample with 50% probability to perform data augmentation. Each training sample size is resized to 112 $\times$ 112. Thus, for 3D convolutions performed on a single-cue input, the input is a video clip with a size of 16 $\times$ 3 $\times$ 112 $\times$ 112 composed of 16 video frames. However, for 3D convolutions performed on a multi-cue input, the input is a new reformed video clip with a size of 16 $\times$ 6 $\times$ 112 $\times$ 112 composed of 16 triple video frames. For the validation process, we uniformly split the video into three video clips to perform data augmentation, and each video clip represents the same video with the same class label.

4.2.3 Recognition process

After training our proposed M3D model and R-M3D model, we use them to recognize the actions in the test dataset. As action recognition is a classification problem [60], classification accuracy is adopted as the main evaluation metric [14, 36, 38]. We train and test our proposed M3D model and R-M3D model on each training/test split of UCF101 dataset and HMDB51 dataset. The average classification accuracies over all three training/test splits are adopted as the final report results. To evaluate the classification accuracy of action recognition for a test video, we split the test video into non-overlapped video clips with a size of 16. Each video clip is cropped from the centre position and resized into 112 $\times$ 112. We use our trained model to classify each video clip to obtain the probabilities for each class label, and average the probabilities of all the video clips corresponding to the test video for classification [35, 39, 42].

4.3 Analysis of experimental results of the M3D model

In this section, we first set up two sets of ablation comparison experiments, we first compare our proposed M3D model with different input modalities in Sect. 4.3.1. Then, we compare the effects of different numbers of layers of CNN for M3D in Sect. 4.3.2. Finally, we analyse the computation cost of our proposed M3D model with the C3D model in Sect. 4.3.3.

4.3.1 Ablation experiments: different input modalities

To compare the different effects of different inputs, we select different input combinations as different comparison models. For fair comparison, all models use the C3D model as the basic model and the only difference is the input part.

Table 1 The network architecture of the proposed R-M3D (35 layers) in detail is illustrated. A residual building block is illustrated in brackets

Layer name	Layer architecture	Output size
Conv1	$[7 \times 7 \times 7, 64]$, kernel size: $7 \times 7 \times 7$, number of feature maps: 64, stride size: $2 \times 2 \times 2$,padding size: $3 \times 3 \times 3$	$56 \times 56 \times 16$, feature map size: 56×56, number of input frames: 16
Maxpool	kernel size: $3 \times 3 \times 3$,stride size: $2 \times 2 \times 2$, padding size: $1 \times 1 \times 1$	$28 \times 28 \times 8$
Conv2_x	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 3$ for all convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$28 \times 28 \times 8$
Conv3_x	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 4$ for the first convolutional layers: stride size: $2 \times 2 \times 2$ padding size: $1 \times 1 \times 1$, for other convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$14 \times 14 \times 4$
Conv4_x	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 6$ for the first convolutional layers: stride size: $2 \times 2 \times 2$ padding size: $1 \times 1 \times 1$, for other convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$7 \times 7 \times 2$
Conv5_x	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 3$ for the first convolutional layers: stride size: $2 \times 2 \times 2$ padding size: $1 \times 1 \times 1$, for other convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$4 \times 4 \times 1$
Averagepool	kernel size: $4 \times 4 \times 1$, stride size: $1 \times 1 \times 1$ padding size: $0 \times 0 \times 0$	$1 \times 1 \times 1$
fc	512×1024	-
Softmax	$1024 \times \{101 \quad or \quad 51\}$	-

All the comparison models can be summarized as follows: (i) Adopting video frames as a single-cue input, *i.e.* C3D baseline model [49]. (ii) Adopting video frame and optical flow as a two-cue input, we call it T3D-I model for short. (iii) Adopting video frame and motion saliency as a two-cue input, we call it T3D-II model for short. (iv) Adopting video frame, optical flow and motion saliency as a multi-cue input, *i.e.* our proposed M3D model. The corresponding best classification accuracies of different input modalities on the UCF101 and HMDB51 datasets are reported in Table 2. We can get these observations as follows:

– As experimental results on the UCF101 dataset illustrated, our proposed M3D model, the T3D-I model, T3D-II model and C3D model obtain 49.0%, 48.5%, 44.8% and 43.2% classification accuracy on the UCF101 dataset, respectively. The first observation is that two-cue input-based or multi-cue input-based models (T3D-I, T3D-II and M3D) outperform the baseline C3D model because more motion-related cues can be provided in the input level. The second observation we can find that the T3D-I model improves much more than the T3D-II model (5.3% vs. 1.6%). This denotes optical flow contributes much more than motion saliency for action recognition, because motion velocity can provide more detailed motion information (*i.e.* motion velocity including motion speed and motion direction) of each pixel while motion saliency mainly provides salient moving pixel information. Lastly, we find that our proposed M3D model obtains the best performance compared with all the other models. The reason is that it can integrate all the appearance cue, direct motion cue and salient motion cue as a multi-cue input.
– As experimental results on the HMDB51 dataset illustrated, we can also find that our proposed M3D model obtains the best performance of classification accuracy (20.8%), followed by the T3D-I model (20.1%), T3D-II model (17.6%) and C3D model (20.8%) just like on the UCF101 dataset. It has been verified that the multi-cue input is more effective than

Table 2 Top-1 classification accuracies of different input modalities of 3D convolution models on UCF101 and HMDB51 datasets (Mean of all accuracies over 3 splits)

Input modalities	UCF101	HMDB51
Video frame (C3D from scratch)	43.2%	16.2%
Video frame + optical flow frame (T3D-I)	48.5%	20.1%
Video frame + motion saliency frame (T3D-II)	44.8%	17.6%
All modalities (M3D)	**49.0%**	**20.8%**

the single-cue input once again and the motion information provided in the input is beneficial for 3D convolutions to improve action recognition. Further, we find the direct motion cue (i.e. motion velocity information) is more important than the salient motion cue (i.e. motion saliency information) for action recognition because the T3D-I model outperforms the T3D-II model (20.1% vs. 17.6%). Our proposed M3D model performs better than the T3D-I model (20.8% vs. 20.1%) because the M3D model provides the extra motion saliency cue in the input level which can suppress background motion noises.

4.3.2 Ablation experiments: different numbers of layers of CNN for M3D

To compare the effectiveness of different numbers of layers, we compare the classification accuracy of different numbers of layers of CNN with different input modalities on UCF101 and HMDB51 in Table 3. According to Table 3, for the input modalities of video frame, i.e. the single-cue input, we can find that the accuracy will improve as the depth of layers increases on both UCF101 and HMDB51 datasets. For the two-cue input (the input modalities of combining video frame with optical flow frame, or the input modalities of combining video frame with motion saliency frame) or the multi-cue input (the input modalities of combining video frame, optical flow frame and motion saliency frame), the same phenomenon appears, i.e. the accuracy will improve as the depth of layers increases. This verifies that the classification accuracy of our proposed M3D can be improved by increase the depth of layers. Thus, we apply our proposed M3D in a deeper CNN model, i.e. ResNet, and compare it with the state of the art in Sect. 4.4.

4.3.3 Computation cost

We compare the computation costs of different models: C3D, T3D-I, T3D-II and our proposed M3D. Concretely, we train the models (C3D, T3D-I, T3D-II and M3D) for 50 epochs on the UCF101 dataset and use the average training time of an epoch as the computation cost to be recorded. The computation costs of different models (C3D, T3D-I, T3D-II and M3D) are reported in Table 4. We can find that the time cost of the compared models is close, almost at the same order of magnitude. We speculate the reason is that the main difference among different models is the number of the input channels and others are the same. Thus the computation cost should be close for different compared models, which demonstrates our proposed M3D can improve the performance of action recognition with the almost same training computation cost compared with C3D. Note that for our proposed M3D, we need extra computation cost for computing the optical flow and motion saliency from the stacked RGB frames in the preprocessing stage.

In addition, we also demonstrate the training convergence of the loss function of the single-cue 3D model (C3D) and the multi-cue 3D model (our proposed M3D) on the UCF101 dataset in Fig. 9. This shows that our proposed M3D model converges faster than the C3D model, which indicates the multi-cue input is more helpful for speeding up model training compared with the single-cue input.

Table 3 Classification accuracy of different numbers of layers of CNN with different input modalities on UCF101 and HMDB51 datasets

Depth of layers of CNN	UCF101	HMDB51
Input modalities: video frame		
2-layer	39.3%	14.3%
4-layer	41.5%	15.0%
6-layer	42.6%	15.4%
Input modalities: video frame + optical flow frame		
2-layer	44.5%	16.2%
4-layer	45.3%	18.5%
6-layer	47.8%	19.2%
Input modalities: video frame + motion saliency frame		
2-layer	41.7%	14.8%
4-layer	43.5%	16.4%
6-layer	44.2%	17.2%
Input modalities: video frame + optical flow frame +motion saliency frame		
2-layer	45.5%	16.1%
4-layer	47.6%	18.6%
6-layer	48.4%	19.4%

Table 4 The computation costs of different models (C3D, T3D-I, T3D-II and M3D) on the UCF101 dataset

Models with different input modalities	Computation cost (s)
C3D (RGB)	1914
T3D-I (RGB + Optical flow)	2044
3D-II (RGB + Motion saliency)	2011
M3D (All modalities)	**2054**

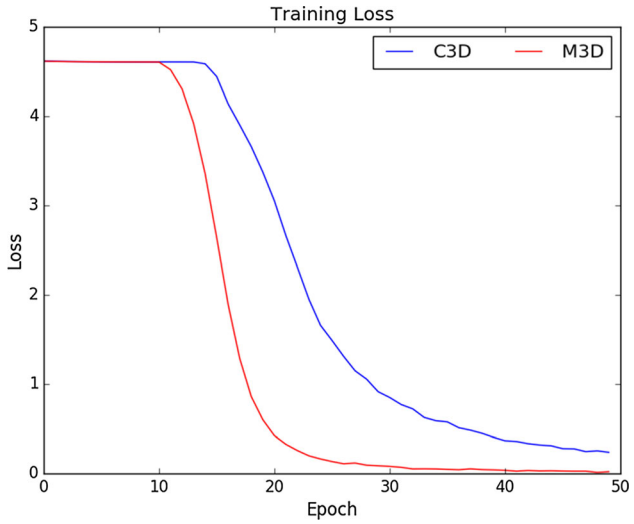

Fig. 9 The training loss of C3D and M3D on the UCF101 dataset, the number of epochs is 50

4.4 Comparing the deep R-M3D model with the state of the art

Based on the above experimental discussion, it has been verified the fact that 3D convolutions performed on a multi-cue input outperform 3D convolutions performed on a single-cue input. However, the classification accuracies are not satisfactory compared with current state of the art models. One reason is that C3D model only contains about 8 layers, which is not deep enough. Another reason is that the scale of UCF101 and HMDB51 datasets only is about 10,000, which are not enough for training a deep neural network. Kinetics dataset [26] is a huge trimmed action dataset, which contains about 300,000 videos and 400 action categories. To explore a deep network on a larger dataset, we fine-tune the proposed R-M3D based on the pretrained Kinetics 3D ResNet on UCF101 and HMDB51 datasets.

We compare our proposed R-M3D ResNet model with other state of the art models including one traditional hand-crafted model (i.e. improved dense trajectories (iDT) [54]) and six deep learning models (i.e. two-stream convolutional networks [45], two-stream with LSTM [57], long-term recurrent convolutional networks (LRCN) [9], long-term temporal convolutions networks (LTCN) [52], Spatiotemporal Residual Networks (ST-ResNet) [10], Pseudo-3D Residual Networks (P3D ResNet) [43], 3D convolutional networks (C3D) [49])and 3D ResNet [17]). Note that 3D ResNet in [17] also is fine-tuned on UCF101 and HMDB51 datasets by using the Kinectics pretrained 3D ResNet. Experimental results on UCF101 and HMDB51 datasets are summarized in Table 5.

According to Table 5, we can find that the proposed fine-tuned R-M3D model outperforms all other compared state of the art models (including iDT+FV, Two-stream networks, Two-stream + LSTM, LRCN, LTCN, C3D (3 nets), P3D ResNet and 3D ResNet) except ST-ResNet on UCF101 and HMDB51 datasets. Especially compared to 3D ResNet, the proposed R-M3D improves 3.9% and 4.4% on UCF101 and HMDB51 datasets, respectively. This demonstrates that 3D convolutions performed on a multi-cue input can improve the action recognition compared with 3D convolutions performed on a single-cue input. We can also find that fine-tuned R-M3D has a significant improvement compared with the M3D model on UCF101 and HMDB51 datasets. This indicates that both a deep network and a larger dataset are important and critical for 3D CNNs to improve action recognition. In addition, we find that the accuracy of our proposed R-M3D is lower 0.2% and 1.0% than ST-ResNet on both UCF101 and HMDB51 datasets, respectively. This reason may be attributed to ST-ResNet can make full use of the complementary information between the two streams of the two-stream architecture.

Table 5 Top-1 classification accuracies of the proposed R-M3D model compared with the state of the art models on UCF101 and HMDB51 datasets (Mean of all accuracies over 3 splits)

Models	UCF101	HMDB51
iDT+FV [54]	85.9%	57.2%
Two-stream networks [45]	86.9%	58.0%
Two-stream + LSTM [57]	88.6%	–
LRCN [9]	82.9%	–
LTCN [52]	91.7%	64.8%
C3D (3 nets) [49]	85.2%	–
P3D ResNet [43]	88.6%	–
ST-ResNet [10]	93.4%	66.4%
3D ResNet (fine-tuned) [17]	89.3%	61.0%
R-M3D (fine-tuned)	**93.2%**	**65.4%**

5 Conclusion

In this paper, we propose a novel M3D model for action recognition. The M3D model directly performs 3D convolutions on a multi-cue input, i.e. stacked triple video frames including appearance information, direct motion information and salient motion information. Compared with the existing 3D CNN-based action recognition methods, the proposed novel triple video representation can integrate the appearance cue, direct motion cue and salient motion cue as input for 3D convolutions. Further, the salient motion cue is robust to background motion noises such as slight leaf jittering and water rippling, which has not been applied in action recognition before. We also develop R-M3D based on the deep 3D ResNet for action recognition. Experimental results verified the effectiveness of our proposed M3D model, and the proposed R-M3D model achieves competitive performance compared with the state-of-the-art.

For future work, in our opinion, the choice of motion cue input is important. In the proposed M3D and R-M3D models, we use two motion detection techniques (i.e. optical flow and motion saliency detection) to capture different motion information. Maybe other approaches can provide more suitable motion information than them. For 3D convolutions performed on a multi-cue input, we develop the triple video representation. If more suitable multi-cue representation methods are developed, we may get even better results.

Acknowledgements This work was in part Supported by the National Key Research and Development Program of China (No. 2018YFB1404102), the Fundamental Research Funds for the Central Universities (No. 2002B02181), Natural Science Foundation of China 51979085, Natural Science Foundation of Jiangsu Province BK2020022539, Major Basic Research of Shandong Natural Science Foundation (ZR2019ZD10), Key Research and Development Plan of Shandong Province (2019GGX101050), Major agricultural application technology innovation project of Shandong Province(SD2019NJ007), China Scholarship Council (CSC) and the New Zealand China Doctoral Research Scholarships Program. Finally, we also thanks to Professor Chunhua Shen and anonymous reviewers for their constructive comments, which significantly improve the quality of this paper.

References

1. Arunnehru J, Chamundeeswari G, Prasanna Bharathi S (2018) Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos. Procedia Computer Sci 133:471–477
2. Baccouche M, Mamalet F, Wolf C, Garcia C, Baskurt A (2011) Sequential deep learning for human action recognition. In: International workshop on human behavior understanding, pp 29–39. Springer, Berlin
3. Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: European conference on computer vision, pp 25–36. Springer, Berlin
4. Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6299–6308
5. Chen Chenglizhao, Li Shuai, Wang Yongguang, Qin Hong, Hao Aimin (2017) Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion. IEEE Trans Image Process 26(7):3156–3170
6. Chen Zhe, Wang Xin, Sun Zhen, Wang Zhijian (2016) Motion saliency detection using a temporal Fourier transform. Opt Laser Technol 80:1–15
7. Cong R, Lei J, Fu H, Cheng MM, Lin W, Huang Q (2018) Review of visual saliency detection with comprehensive information. IEEE Trans Circuits Syst Video Technol 29:2941
8. Cui X, Liu Q, Zhang S, Yang F, Metaxas DN (2012) Temporal spectral residual for fast salient motion detection. Neurocomputing 86:24–32
9. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2625–2634
10. Feichtenhofer C, Pinz A, Wildes RP (2016) Spatiotemporal residual networks for video action recognition. In: Advances in neural information processing systems, pp 3468–3476
11. Feichtenhofer C, Pinz A, Wildes RP (2017) Spatiotemporal multiplier networks for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4768–4777
12. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1933–1941
13. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
14. Gong W, Qi L, Xu Y (2018) Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wireless Commun Mobile Comput. https://doi.org/10.1155/2018/3075849
15. Guo C, Ma Q, Zhang L (2008) Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: IEEE conference on computer vision and pattern recognition, pp 1–8. IEEE
16. Hara K, Kataoka H, Satoh Y. (2017) Learning spatio-temporal features with 3d residual networks for action recognition. In: Proceedings of the IEEE international conference on computer vision, pp 3154–3160
17. Hara K, Kataoka H, Satoh Y (2018) Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6546–6555
18. Harel J, Koch C, Perona P (2007) Graph-based visual saliency. In: Advances in neural information processing systems, pp 545–552
19. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
20. Horn Berthold KP, Schunck Brian G (1981) Determining optical flow. Artif Intell 17(1–3):185–203
21. Hou X, Zhang L (2007) Saliency detection: a spectral residual approach. In: IEEE conference on computer vision and pattern recognition, pp 1–8. IEEE

22. Itti Laurent, Koch Christof, Niebur Ernst (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell 11:1254–1259
23. Ji Shuiwang, Wei Xu, Yang Ming, Kai Yu (2013) 3d convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
24. Ji Yuzhu, Zhang Haijun, Wu QM Jonathan (2018) Salient object detection via multi-scale attention cnn. Neurocomputing 322:130–140
25. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1725–1732
26. Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Trevor B, Paul N et al 2017 The kinetics human action video dataset. arXiv preprint arXiv:1705.06950
27. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
28. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) Hmdb: a large video database for human motion recognition. In: 2011 International conference on computer vision, pp 2556–2563
29. Liu Z, Li Z, Wang R, Zong M, Ji W (2020) Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05144-7
30. Liu Z, Li Z, Zong M, Ji W, Wang R, Tian Y (2019) Spatiotemporal saliency based multi-stream networks for action recognition. In: Asian conference on pattern recognition, pp 74–84. Springer, Singapore
31. Shelhamer E, Long J, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
32. Pereira Eduardo M, Ciobanu Lucian, Cardoso Jaime S (2017) Cross-layer classification framework for automatic social behavioural analysis in surveillance scenario. Neural Comput Appl 28(9):2425–2444
33. Pérez JS, Meinhardt-Llopis E, Facciolo G (2013) TV-L1 optical flow estimation. Image Process On Line 2013:137–150
34. Qi L, Chen Y, Yuan Y, Fu S, Zhang X, Xu X (2019) A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. World Wide Web 23:1275
35. Qi Lianyong, Dai Peiqiang, Jiguo Yu, Zhou Zhili, Yanwei Xu (2017) Time-location-frequency-aware internet of things service selection based on historical records. Int J Distrib Sens Netw 13(1):1550147716688696
36. Qi Lianyong, Dou Wanchun, Chen Jinjun (2016) Weighted principal component analysis-based service selection method for multimedia services in cloud. Computing 98(1–2):195–214
37. Qi Lianyong, Wang Ruili, Chunhua Hu, Li Shancang, He Qiang, Xiaolong Xu (2019) Time-aware distributed service recommendation with privacy-preservation. Inf Sci 480:354–364
38. Qi L, Xu X, Dou W, Yu J, Zhou Z, Zhang X (2016) Time-aware IOE service recommendation on sparse data. Mobile Inf Syst 2016:4397061. https://doi.org/10.1155/2016/4397061
39. Qi L, Yu J, Zhou Z (2017) An invocation cost optimization method for web services in cloud environment. Scientific Program. https://doi.org/10.1155/2017/4358536
40. Qi Lianyong, Zhang Xuyun, Dou Wanchun, Chunhua Hu, Yang Chi, Chen Jinjun (2018) A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. Future Gener Comput Syst 88:636–643
41. Qi Lianyong, Zhang Xuyun, Dou Wanchun, Ni Qiang (2017) A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data. IEEE J Sel Areas Commun 35(11):2616–2624
42. Qi Lianyong, Zhou Zhili, Jiguo Yu, Liu Qi (2017) Data-sparsity tolerant web service recommendation approach based on improved collaborative filtering. IEICE Trans Inf Syst 100(9):2092–2099
43. Qiu Z, Yao T, Mei T (2017) Learning spatio-temporal representation with pseudo-3d residual networks. In: Proceedings of the IEEE international conference on computer vision, pp 5533–5541
44. Shamsolmoali Pourya, Zareapoor Masoumeh, Wang Ruili, Jain Deepak Kumar, Yang Jie (2019) G-ganisr: gradual generative adversarial network for image super resolution. Neurocomputing 366:140–153
45. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
46. Soomro K, Zamir AR, Shah M (2012) Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402
47. Tian Chunwei, Yong Xu, Zuo Wangmeng (2020) Image denoising using deep cnn with batch renormalization. Neural Netw 121:461–473
48. Tian C, Xu Y, Zuo W, Zhang B, Fei L, Lin CW (2020) Coarse-to-fine CNN for image super-resolution. IEEE Trans Multimedia. https://doi.org/10.1109/TMM.2020.2999182
49. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4489–4497
50. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6450–6459
51. Tu Z, Li H, Zhang D, Dauwels J, Li B, Yuan J (2019) Action-stage emphasized spatio-temporal vlad for video action recognition. IEEE Trans Image Process 28:2799
52. Varol Gül, Laptev Ivan, Schmid Cordelia (2018) Long-term temporal convolutions for action recognition. IEEE Trans Pattern Anal Mach Intell 40(6):1510–1517
53. Wang H, Kläser A, Schmid C, Liu CL, (2011) Action recognition by dense trajectories. In: IEEE conference on computer vision & pattern recognition, pp 3169–3176
54. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision, pp 3551–3558
55. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: towards good practices for deep action recognition. In: European conference on computer vision, pp 20–36. Springer, Singapore
56. Xue Y, Guo X, Cao X (2012) Motion saliency detection using low-rank and sparse decomposition. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 1485–1488
57. Ng YH Joe, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: deep networks for video classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4694–4702
58. Zeng Shaoning, Gou Jianping, Yang Xiong (2018) Improving sparsity of coefficients for robust sparse and collaborative representation-based image classification. Neural Comput Appl 30(10):2965–2978

59. Zhang Haijun, Ji Yuzhu, Huang Wang, Liu Linlin (2019) Sitcom-star-based clothing retrieval for video advertising: a deep learning framework. Neural Comput Appl 31(11):7361–7380
60. Zhang Shichao, Li Xuelong, Zong Ming, Zhu Xiaofeng, Wang Ruili (2018) Efficient knn classification with different numbers of nearest neighbors. IEEE Trans Neural Netw Learn Syst 29(5):1774–1785
61. Zheng H, Wang R, Ji W, Zong M, Wong WK, Lai Z, Lv H (2020) Discriminative deep multi-task learning for facial expression recognition. Inf Sci. https://doi.org/10.1016/j.ins.2020.04.041
62. Zhou Y, Sun X, Zha ZJ, Zeng W (2018) Mict: Mixed 3d/2d convolutional tube for human action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 449–458