Multimed Tools Appl
https://doi.org/10.1007/s11042-018-5893-9

Human action recognition based on quaternion spatial-temporal convolutional neural network and LSTM in RGB videos

Bo Meng[1] · **XueJun Liu**[1] · **Xiaolin Wang**[1]

Received: 8 June 2017 / Revised: 31 January 2018 / Accepted: 13 March 2018

Abstract Convolutional neural networks (CNN) are the state-of-the-art method for action recognition in various kinds of datasets. However, most existing CNN models are based on lower-level handcrafted features from gray or RGB image sequences from small datasets, which are incapable of being generalized for application to various realistic scenarios. Therefore, we propose a new deep learning network for action recognition that integrates quaternion spatial-temporal convolutional neural network (QST-CNN) and Long Short-Term Memory network (LSTM), called QST-CNN-LSTM. Unlike a traditional CNN, the input for a QST-CNN utilizes a quaternion expression for an RGB image, and the values of the red, green, and blue channels are considered simultaneously as a whole in a spatial convolutional layer, avoiding the loss of spatial features. Because the raw images in video datasets are large and have background redundancy, we pre-extract key motion regions from RGB videos using an improved codebook algorithm. Furthermore, the QST-CNN is combined with LSTM for capturing the dependencies between different video clips. Experiments demonstrate that QST-CNN-LSTM is effective for improving recognition rates in the Weizmann, UCF sports, and UCF11 datasets.

Keywords Human action recognition · Convolutional neural network · Quaternion · Long short-term memory network · Codebook

1 Introduction

Action recognition is a research focus point in the computer vision field and is widely used in many other fields, including human-computer interaction, intelligent surveillance systems, and home security [11, 28, 35]. However, compared to still image classification, human action is

✉ XueJun Liu
liuxuejun_0828@163.com

[1] School of Information Engineering, Northeast Electric Power University, Jilin, China

Published online: 22 March 2018

recognized as requiring a complicated procedure that includes spatial and temporal information from videos. Thus, effectively extracting video features for action recognition is still considered a challenging problem.

It was shown in [26] that methods of extracting features can be divided into two categories: handcrafted feature methods, and automatic learning feature methods. Handcrafted action features, such as histograms of optical flows (HOF) [3], histograms of oriented gradients (HOG) [5, 46], and space-time interest points (STIP) [23, 31], are extracted from video sequences. This has been best practice for image and video classification until recently. Improved Dense Trajectories (IDT) [40], based on speeded up robust features (SURF) and dense optical flows, have become the most popularized action representations among all handcrafted approaches. The reason behind this popularity is that handcrafted features are carefully designed for a specific dataset, and do not rely on any labeled data. However, handcrafted features are usually ad-hoc, and cannot be generalized for application to a broader variety of realistic scenarios.

There have been several recent attempts to develop deep learning architectures for automatically learning features, especially CNN. 2D image-based CNNs, such as deep Convolutional Networks (ConvNets) [18], have been extended to 3D CNNs for learning spatial and temporal features from video sequences. However, most proposed CNN architectures still rely on lower-level handcrafted features. A 3D CNN architecture, consisting of seven layers for action recognition, was proposed in [15]. It uses 5 different features known as gray, gradient-x, gradient-y, optical flow-x, and optical flow-y. Optical flow features are widely used to represent dynamic information for human actions, such as two-stream CNNs [2], three-stream CNNs [43], and pose-based CNNs (P-CNN) [4]. There are also CNN-based frameworks to automatically learn a spatial-temporal feature from a raw RGB video sequence. However, those networks are deeper, and too complex to learn the accurate high-level representations of human action in larger datasets.

In this paper, we focus on how to improve the feature extraction performance of a CNN without dependence on the lower-level handcrafted features. The quaternion, which was proposed by Hamilton in 1843 [12], is a hypercomplex number. The quaternion representation of color images has been successfully applied in many computer vision fields, such as face recognition [42, 48], and image classification [19, 47]. In [45], compared with feature extraction from gray images, some features including edge intensity and average gradient are improved by quaternion operation on color images. However, those spatial features are also important for action recognition; for example, strengthening edge intensity is essential when the color of clothes is similar to the background color.

Based on the aforementioned problems, we present a quaternion spatial-temporal convolutional neural network for human action recognition in RGB videos. The main contributions of the proposed approach are follows:

(1) The proposed QST-CNN considers the spatial distribution of information in RGB channels by using quaternions in its spatial convolutional layer, and considers the dynamical information of adjacent frames in its temporal convolutional layer. It enhances the spatial features of color images by integrating the values of the RGB channels into its convolutional operations.
(2) Because the raw images in video datasets are large and have background redundancy, the key motion regions in RGB videos are extracted using an improved codebook algorithm.

(3) The inputs for QST-CNN are video clips, rather than whole videos, which tend to lose the connections between different video segments. Thus, we use an LSTM to capture the dynamical dependencies between video clips. Our model achieves good performance for action recognition in the Weizmann, UCF sports, and UCF11 datasets.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes our method for action recognition. Section 4 presents experimental evaluations. Finally, we provide conclusions and discuss future work in Section 5.

2 Related work

2.1 Handcrafted feature-based action recognition

Handcrafted feature-based action representations have dominated the computer vision field alongside action recognition [9, 39]. Before the emergence of deep learning approaches, most of the handcrafted feature methods used in action recognition employed a fixed procedure, including feature extraction, feature representations, and action classification.

In this approach, low-level action features, such as HOF, HOG, and STIP, are extracted from those sparse spatial-temporal features, and projected to video-level representations through coding methods, such as the bag-of-words (BOW) model [38], and sparse coding (SC) [1]. However, some later works prove that improved performance can be achieved by adopting densely sampled spatial-temporal features. Wang et al. [40] proposed an action recognition method with improved dense trajectories (IDT) that integrates SURF and optical flows. Their method improved motion-based descriptors significantly. Motivated by the success of IDT, researchers are working intensely towards developing IDT for video-feature learning. In [32], the performance of IDT is enhanced by building a multi-layer stacked Fisher Vector (FV) method. The authors of [36] proposed a method to sub-sample and generate vocabularies for DT features. In [13], the performance of base action classifiers is improved by adopting three methods: data augmentation, Subsequence-Score Distribution, and Least-Squares SVMs. The authors of [21] proposed an approach called Multi-skip Feature Stacking (MIFS), and achieved state-of-the-art results on several datasets. In 2016, Lan et al. [22] proposed leverage effective techniques from both data-driven and data-independent approaches, to improve action recognition systems. However, although the extraction of more spatial-temporal features from the video improves performance, it also incurs higher computational costs.

2.2 Deep learning-based action recognition

Deep learning with learned parameters has been proposed to automatically learn features using layer-wise training methods. CNN is widely used in computer vision field, including object recognition, image semantic retrieval, and image steganography [10].This further extends the use of convolutional networks as generic feature extractors.

Spatial and temporal information is generally extracted from video using the convolutional operation of deep networks. The authors [17] propose a modified convolutional neural network (MCNN)-based action handcrafted features extraction and classification framework, where the outer boundary information is extracted using three-dimensional Gabor filters. A 3D CNN network is introduced in [15], which significantly outperforms the 2D frame-based architecture. There have been attempts to improve CNN, due to its popularity. A feature-learning architecture was established in [24] by combining a CNN with independent subspace analysis (ISA), with the goal of extracting hierarchical invariant spatial-temporal (HIST) features. The authors of [2] later proposed two-stream ConvNets architecture, designed to capture complementary information regarding appearance, from still frames and the motion between frames. In [30], a new approach is proposed for combining different sources of knowledge in Deep CNNs for action recognition, which outperforms methods using only CNNs and optical flow features. To exploit temporal information, some studies resort to the use of recurrent structures. The works of [6, 7, 27] tackle the problem of action recognition through the construction of a Long–Short Term Memory (LSTM) network on top of CNNs, to capture longer-term relationships among frames. All of the above rely on numerous labels, which are expensive to obtain, and generally perform worse than handcrafted features among small datasets.

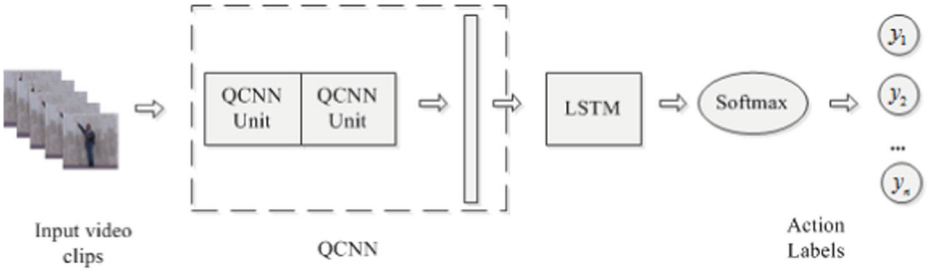
3 Proposed method

In this section, we describe the QST-CNN-LSTM model for action recognition. Fig. 1 presents an overview of the proposed model. The model is constructed with two QST-CNN units, a fully connected layer, and an LSTM. An image sequence, is given to the first QST-CNN unit as input. Color features are then extracted in a quaternion spatial convolutional layer. The temporal convolutional layer and subsampling operation are designed for processing channels separately, where the red, green, blue, and pink boxes denote the red, green, and blue channels, and the color image, respectively. Finally, softmax regression is used to classify the action labels.

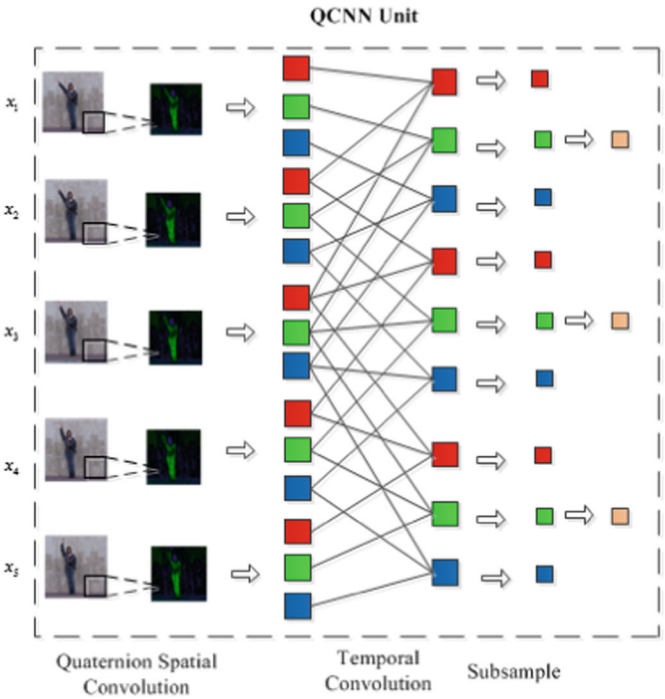
3.1 Key motion region extraction using an improved codebook model

The codebook algorithm was proposed in [16] for constructing background models from long observation sequences. This algorithm consists of two steps: background modeling, and detection. Let $X = \{x_1, x_2, \ldots, x_n\}$ be the coding sequence for a pixel X, where $C = \{c_1, c_2, \ldots, c_L\}$ is a codebook consisting of L codewords for the pixel X. Each codeword includes the average of RGB vectors $v_i = \left(\overline{R}_i, \overline{G}_i, \overline{B}_i\right)$ and a six-tuple array $aux_i = \left(I^{\vee}{}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i\right)$, where $I^{\vee}{}_i$ and $\hat{I}_i$ denote the maximum and minimum of brightness during codeword modeling, respectively. f_i is the frequency of this codeword, p_i and q_i are the first and last match times, and λ_i is the time interval. In the background modeling phase, the brightness criterion is defined as:

$$brightness\left(I, <\overset{\vee}{I}_i, \hat{I}_i, >\right) = \left\{ \begin{array}{ll} true & \alpha\hat{I}_i \le \|x_i\| \le \min\left\{\beta\hat{I}_i, \overset{\vee}{I}_i/\alpha\right\} \\ false & otherwise \end{array} \right\}, \tag{1}$$



(a) Overview of the proposed method.



(b) Architecture of a QST-CNN unit.

Fig. 1 Architecture of QST-CNN-LSTM

where $\alpha(\alpha < 1)$ and $\beta(\beta > 1)$ are the thresholds for the brightness range. Additionally, color distortion is defined as:

$$colordist(x_t, v_i) = \sqrt{\left(R^2 + G^2 + B^2\right) - \frac{\left(\overline{R}_i R + \overline{G}_i G + \overline{B}_i B\right)}{\overline{R}_i^{\,2} + \overline{G}_i^{\,2} + \overline{B}_i^{\,2}}}. \tag{2}$$

Each incoming pixel is classified as a foreground pixel, unless it satisfies two conditions, in which case it is classified as a background pixel. The relevant conditions are whether the color distortion for certain codewords is less than the detection threshold, and whether the brightness lies within the brightness range of the codeword for each pixel.

Due to the complexity and dynamics of a real scene, it is difficult to accurately segment human action regions using the traditional codebook method. We propose a key motion region extraction method, based on frame difference and the codebook algorithm in the RGB space. Frame difference is used to establish approximate regions of human motion in raw images. Accurate segmentation is achieved by using the approximate region video as input for a codebook model. There are three criteria used to determine if the pixel belongs to the background. During background modeling, we use the brightness and color distortion criteria introduced in Eqs. (1) and (2), and the frame difference criterion is defined as:

$$D(x_t) = \begin{cases} true & if\ |x_t - x_{t-1}| > T \\ false & otherwie \end{cases}, \tag{3}$$

where x_t and x_{t-1} denote the pixel value at frames t and $t-1$, and T is a threshold. The details of background modeling are discussed below. For each dataset, a bounding box is used to extract the foreground from the raw images. The key motion region images for the Weizmann, UCF sports, and UCF11 datasets are presented in Fig. 2. They use bounding boxes of size 90×90, 250×400, and 110×125 respectively. Because the diving and lifting action region images are not provided in the UCF sports dataset, we extract them using a bounding box of the same size as that used for other action region images, which are rescaled to 250×400 using the method in [20].

Algorithm for Codebook Construction

I. Initialize a codeword c_L, $L \leftarrow 0$;

II. for $t = 1$ to N, do

(i) Input a pixel $x_t(R,G,B)$, find the codeword c_m in an original pixel matching x_t based on three conditions (a), (b), and (c):

(a) $colordist(x_t, v_m) \le \varepsilon_1$

(b) $brightness(I, \langle \check{I}_m, \hat{I}_m \rangle) = true$

(c) $D(x_t) = true$

(ii) If the codeword is an empty set or there are no matches, then create a new codeword c_L by setting two parameters:

(a) $v_L \leftarrow (R,G,B)$

(b) $aux_L \leftarrow \langle I, I, 1, t-1, t, t \rangle$

(iii) Otherwise, update the matching codeword c_m, $v_m = (\overline{R}_m, \overline{G}_m, \overline{B}_m)$:

(a) $v_m \leftarrow \left(\frac{f_m \overline{R_m} + R}{f_m + 1}, \frac{f_m \overline{G_m} + G}{f_m + 1}, \frac{f_m \overline{G_m} + G}{f_m + 1}\right)$

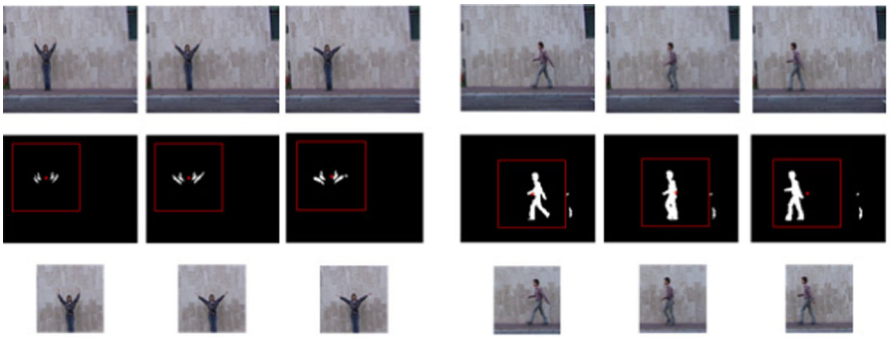
(b) $aux_m \leftarrow \langle \min\left\{I, \check{I}_m\right\}, \max\left\{I, \hat{I}_m\right\}, f_m + 1, \max\left\{\lambda_m, t - q_m\right\}, p_m, t \rangle$

End

III. For each codeword c_i, $i = 1, ..., L$, wrap around $\lambda_i \leftarrow \max\left\{\lambda_i, (N - q_i + p_i - 1)\right\}$.

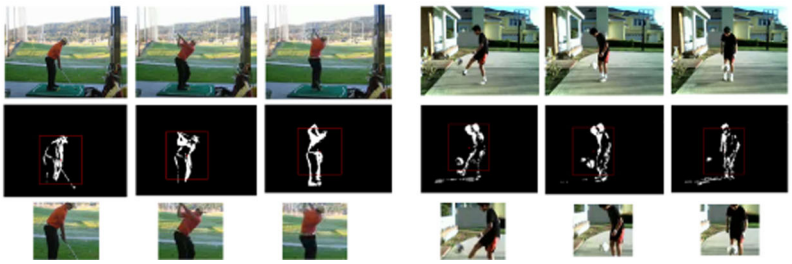
IV. An incoming pixel value x is defined as $I(x)$ in the test:

$$I(x) = \begin{cases} background & if\ there\ is\ match \\ action\ region & otherwise \end{cases}$$



(a) Example action regions extracted from the Weizmann dataset.

(b) Example action regions extracted from the UCF sports dataset.



(c) Example action regions extracted from the UCF11 dataset.

Fig. 2 Example action regions extracted from three datasets

3.2 Quaternion convolutional neural network for human action recognition

3.2.1 Quaternion spatial convolution

In the spatial convolutional layers, 2D spatial convolution kernels, designed for feature extraction from gray images, are expanded into pure quaternions $W = (W_r, W_g, W_b)$. Given an RGB input frame $Q = (Q_r, Q_g, Q_b)$, the value at position (x, y) in the jth feature map of the ith layer is defined as:

$$Z^{i,j}(x,y) = f\left(\sum_{p}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} conv\left(W_{i,j,p}^{n,m}, Q^{(i-1),p}(x+n, y+m)\right) + b^{i,j}\right) \tag{4}$$

$$conv(W, Q) = W \otimes Q + W \times Q \tag{5}$$

$$W \otimes Q = \left(W_r Q_r, W_g Q_g, W_b Q_b\right) \tag{6}$$

$$W \times Q = \left(W_g Q_g, W_b Q_g, W_b Q_r - W_r Q_b, W_r Q_g, W_g Q_r\right). \tag{7}$$

In Eq. (4), f is sigmoid function, $b^{i,j}$ is the bias, $W_{i,j,p}^{n,m}$ is the vector at position (n, m) of the kernel connected to the p th feature map of the (i-1) th layer, and N and M are the height and width of the feature map. Equation (5) is the convolutional operation between the image and the kernel. The operator $\otimes$ is defined as the element-wise product of vectors, and the operator $\times$ denotes the cross product of two pure quaternions. The quaternion spatial convolution considers not only the spatial interactions within each of the three color channels (Eq. (6)), but also the interaction between different color channels (Eq. (7)).

3.2.2 Temporal convolution and subsampling

Different from pose recognition, action recognition is the process of modeling human action patterns from videos instead of single images. The motion information encoded in multiple adjacent frames is crucial. Given the output feature map sequence $Z = \{Z_1, Z_2, \ldots, Z_t\}$ from quaternion spatial convolution, the temporal convolution of each channel is:

$$z_{c,t}^{i} = f\left(\sum_{s=0}^{S} w_{c,s}^{i} z_{c,t+s}^{i-1} + b^{i}\right), \tag{8}$$

where $w_{c,s}^{i}$ represents the weight of s th temporal scale of the cth channel and $z_{c,t+s}^{i-1}$ denotes the value of the c th channel. Given an input sequence of length T, the size of the output sequence is $T^{i} = T - S + 1$. The subsampling layer, which uses an averaging operation on each channel, is performed after the temporal convolutional layer.

3.2.3 Architecture of the network

As show in Fig. 3, the architecture for action recognition is constructed using a combination of a QST-CNN and an LSTM. The QST-CNN consists of two QST-CNN units and a fully-connected layer. Each QST-CNN unit includes a quaternion spatial convolutional layer, a temporal convolutional layer, and a subsampling layer. The lengths of action videos and the scale of images may vary for different datasets, so different datasets have different sizes of video clips as inputs for the QST-CNN. Using the Weizmann dataset as an example, a

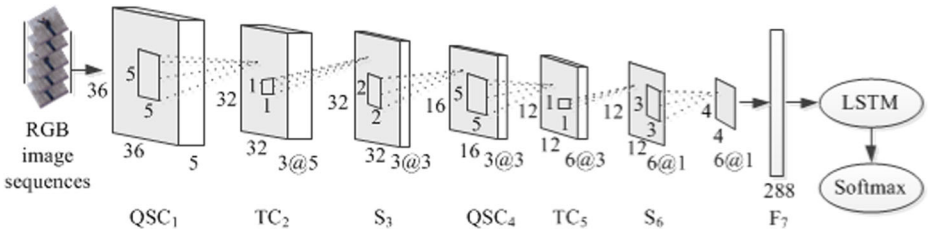


Fig. 3 Architecture of QST-CNN

bounding box of size 90 × 90 is used to extract the key region for the codebook model. An input image of size 36 × 36 is then acquired by using the Nearest Neighbor algorithm. Firstly, an RGB video clip of size 36 × 36 × 5 (height×width×frames) is used as input for the network. We then apply two quaternion convolutional layers (QSC_1 and QSC_4) with a kernel of size 5 × 5 × 3 (height×width×channels), with kernel numbers of 3 and 6, respectively. Two temporal convolutional layers (TC_1 and TC_5) of scale 3, as well as a 2 × 2 and 3 × 3 subsampling operation (S_1 and S_5), are applied following the quaternion convolutional layers. Finally, 96 pure quaternions are sent to the fully-connected layer. They are converted into a 288-dimensional vector, which is used as the input for the LSTM.

3.3 Model implementation with long short-term memory network

After the features have been extracted by the QST-CNN, the model for high-level action recognition is implemented with an LSTM, which enables temporal dynamics for action videos. In this model, the LSTM is applied in its most general form [14], consisting of an input unit i_t, an input modulation unit m_t, a forget unit f_t, a memory unit c_t, and an output unit o_t. The LSTM updates as follows:

$$i_t = s(w_{xi}x_t + w_{hi}x_{t-1} + b_i) \tag{9}$$

$$f_t = s\left(w_{xf}x_t + w_{hf}x_{t-1} + b_f\right) \tag{10}$$

$$o_t = s(w_{xo}x_t + w_{ho}x_{t-1} + b_o) \tag{11}$$

$$m_t = \tanh(w_{xc}x_t + w_{hc}x_{t-1} + b_c) \tag{12}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes m_t \tag{13}$$

$$h_t = o_t \otimes \tanh(c_t), \tag{14}$$

where s is sigmoid function, w and b are the model parameters, x_t is a fixed-length output vector from the QST-CNN, and h_t denotes the state of the LSTM at time step t. Because the sigmoid function transforms input values into a [0, 1] range, i_t, f_t, and o_t are used as a gate that enables the LSTM to selectively forget its previous sequence features or retain them as an input unit, forget unit, and output unit. The output distribution of an action sequence is predicted by a softmax function, where the maximum probability denotes the action label for an action sequence.

4 Experimental results

We evaluate our approach on three publicly available datasets Weizmann [8], UCF sports [34], and UCF11 [25].We compare the performance against the baseline network gray single

channel CNN (Gray-CNN) and the extended network RGB three channel CNN (3Channel-CNN), which provide a performance reference for QST-CNN-LSTM network. The performance has also been compared with state-of-the-art methods using CNN and LSTM for those three databases. The proposed framework is based on the deep learning toolbox [29], available in MATLAB R2015a.

4.1 Datasets

Weizmann dataset includes ten actions: *bending, jacking (jumping-jack), jumping (jumping forward on two legs), pjumping (jumping in place on two legs), running, siding, skipping, walking, waving1 (waving one hand),* and *waving2 (waving two hands)*, with each action being performed by nine subjects. We follow the experimental setup suggested in [8], where video sequences for five people are selected randomly for training, and the others are used for testing.

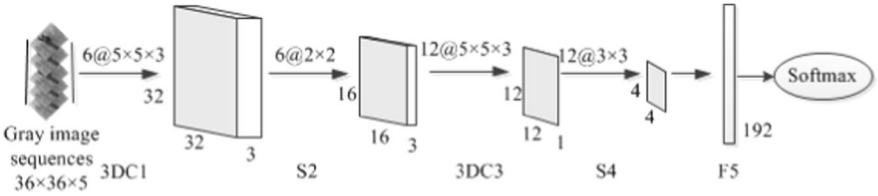
UCF sports dataset is comprised of ten activities: *diving, golf swing, kicking, lifting, riding horse, running, skateboarding, swing-bench, swing-side,* and *walking*. It was collected from various sports on broadcast television channels, such as the BBC and ESPN. The human actions in the UCF sports dataset are more complicated than those in the Weizmann dataset. We follow the aforementioned cross-subject validation experimental setup, where half of the videos are used for training and the other half for testing.

UCF11 is dataset the YouTube Action dataset consisting of 1600 videos and 11 actions: *basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog*. The dataset is considered a challenging dataset because of its large variation of illumination conditions, camera motion, and cluttered backgrounds, which are present within the video sequences. We use 975 videos for training and 625 videos for testing.

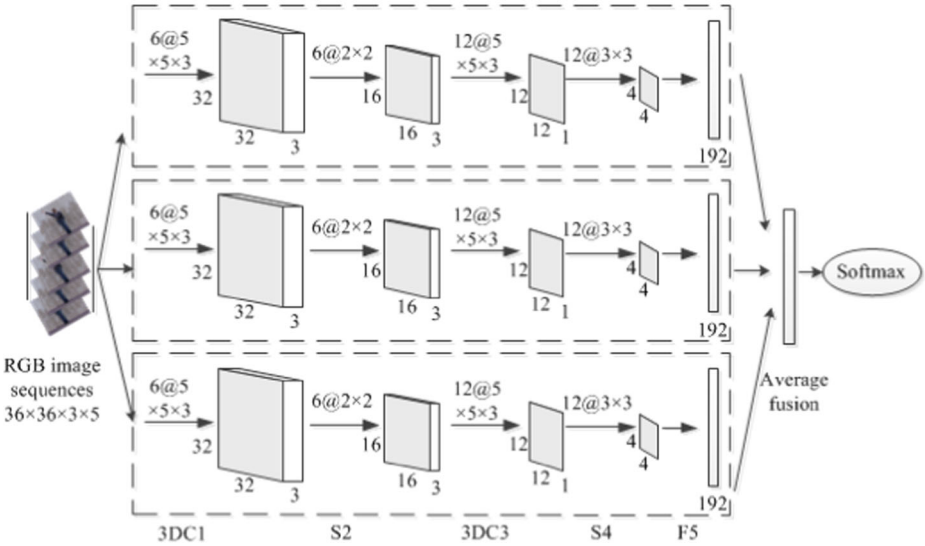
4.2 Results on Weizmann dataset

To evaluate our approach without lower-level handcrafted features, we construct 3D CNN model introduced in [15], based on the Weizmann dataset. Unlike [15], the input for Gray-CNN utilizes raw gray image sequence not gradient and optical flow features. Additionally, 3Channel-CNN is used to evaluate the effect of color channel on feature extraction. Gray-CNN consists of five layers, as shown in Fig. 4a. First, gray sequences of size $36 \times 36 \times 5$ (height×width×frames) are input to the network, then convolutional kernels of size $5 \times 5 \times 3$ are applied in two 3D convolutional layers, with kernel numbers of 6 and 12, respectively. Two subsampling operations, 2×2 and 3×3, are applied in two average pooling layers, resulting in a 192-dimensional vector in the fully-connected layer. As shown in Fig. 4b, 3Channel-CNN consists of three Gray-CNNs, where the action class scores are the average of the output from the three channels. QST-CNN-LSTM is described in Section 3. Note that the learning rates are set to 0.1, the batch size is 50, the number of iterations is 10, and the Dropout rate is 0.4.

Table 1 contains the performance comparison results for different recognition methods. Each test was repeated 20 times, and the maximum performance is shown in Table 1. The results show that the QST-CNN-LSTM significantly outperforms the other three models. One can see that QST-CNN-LSTM achieved 96.34% recognition accuracy, whereas the accuracies of 3D CNN, Gray-CNN, and 3Channel-CNN were 90.12%, 86.46%, and 76.00%, respectively. Experiments also demonstrate that the QST-CNN-LSTM is effective for improving



(a) Architecture of Gray-CNN.



(b) Architecture of 3Channel-CNN.

Fig. 4 Architectures of Gray-CNN and 3Channel-CNN. (a) Gray-CNN. (b) 3Channel-CNN

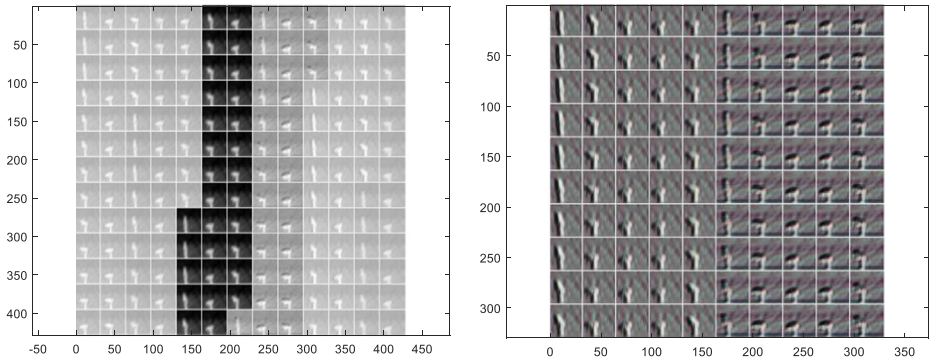
recognition rates without low-level features. Figure 5 presents the output feature maps from the first convolutional layer in Gray-CNN and QST-CNN-LSTM, with human features being more obvious in the latter model. To overcome the shortcomings of subjective evaluation, this paper uses average gradient, edge intensity, and entropy of information to evaluate the effect of image feature extraction. As listed in Table 2, feature values extracted by QST-CNN-LSTM are higher than those extracted by Gray-CNN, which shows that QST-CNN-LSTM has obvious advantages in feature extraction of color images.

4.3 Results on UCF sports dataset

An input image of size 400×250 is reduced to 80×50 by using the nearest neighbor algorithm. First, an RGB video clip of size $80 \times 50 \times 7$ (height×width×frames) is input to a

Table 1 Recognition accuracy (%) on the Weizmann dataset

Method	Accuracy (%)
3Channel-CNN	76.00
Gray-CNN	86.46
3D CNN[15]	90.12
QST-CNN-LSTM	96.34



(a) Gray-CNN (b) QST-CNN-LSTM

Fig. 5 Visualization of example feature maps from the first convolutional layer. (a) Gray-CNN, (b) QST-CNN-LSTM

QST-CNN-LSTM. We then apply three quaternion convolutional layers with a kernel of size $7 \times 7 \times 3$, $5 \times 5 \times 3$, $5 \times 5 \times 3$ (height×width×channels) and kernel numbers of 3, 6, and 12. Three temporal convolutional layers of scale 3, and two subsampling operations of size 2×2 and 3×3, are applied after the quaternion convolutional layers. Finally, 168 pure quaternions are calculated in the fully-connected layer as the input for the LSTM. In the experiment, the learning rate is set to 0.1, the batch size is 50, the number of iterations is 10, and the Dropout rate is 0.5.

The confusion matrix for the proposed method is presented in Fig. 6. Seven actions are classified more than 90% correctly for this dataset. However, some confusion occurs between pairs of different actions, such as "golf swing" and "walking," and "kicking" and "swing-bench." The lowest accuracy achieved was for the action "riding horse" (86%). One can examine the confusion between the actions "walking" "golf swing" and "kicking", to further analyze the advantages of the QST-CNN-LSTM. Twenty video clips for each action are randomly selected and the action probability distribution of each video clip is plotted for the videos of the three actions used for testing. In Fig. 7, video clips one to twenty are of the action "walking," video clips twenty-one to forty are of the action "golf swing" and video clips forty-one to sixty are of the action "kicking" in the labeled test set. In comparison with Fig. 7(a) and (b), the curve peak of each action is closer to the label value (1) in its own range in Fig. 7(c). The QST-CNN-LSTM model effectively improves the probability of correct action identification, and reduces the probability of other actions being identified. Furthermore, the confusion in video clips thirty to forty is reduced using the QST-CNN-LSTM, proving that the QST-CNN-LSTM performs well in recognizing the correct actions.

Table 3 contains the performance comparison for the different recognition methods. Results show that the QST-CNN-LSTM consistently outperforms the baseline networks Gray-CNN and 3Channels-CNN for this dataset. The QST-CNN-LSTM offers better results than the

Table 2 Objective evaluation of feature map

Method	Average Gradient	Edge Intensity	Entropy of Information
Gray-CNN	3.54	35.08	5.51
QST-CNN-LSTM	4.37	35.92	6.47

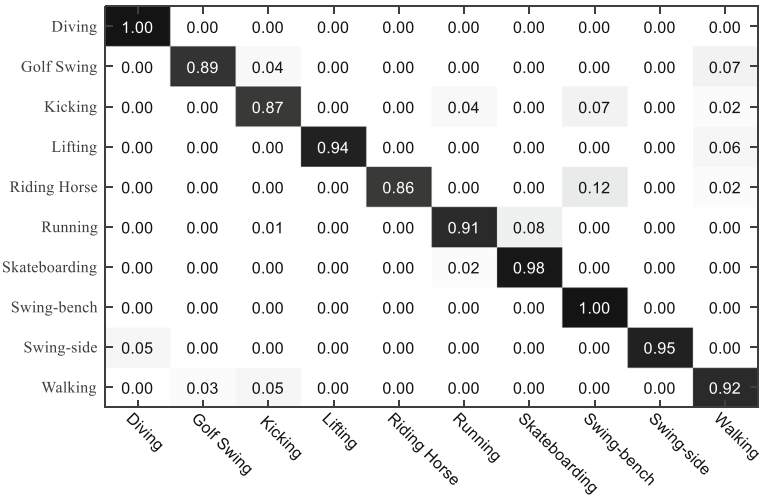
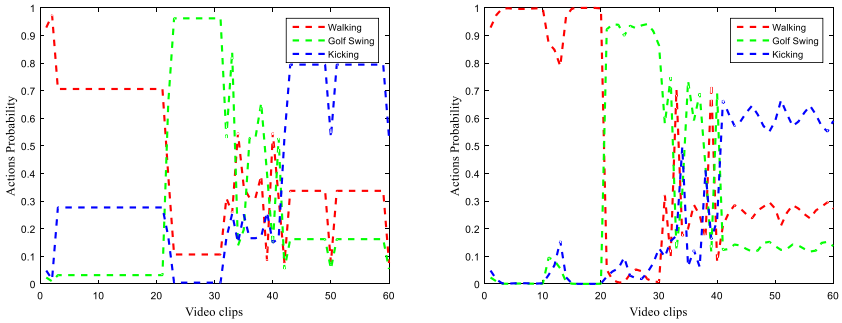
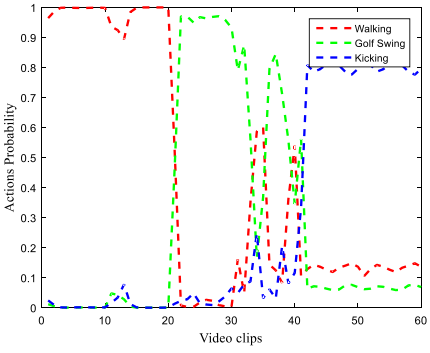


Fig. 6 Confusion matrix for the proposed method for the UCF sports dataset

Dense Trajectories in the recognition of models. Dense Trajectories [41] is the most popularized action representations among all handcrafted approaches. Among deep architectures, QST-CNN-LSTM systematically performs better than the other CNNs and LSTM, improving the performance by 1% on the UCF sports dataset.



(a) Probabilities of three actions in 3Channel-CNN. (b) Probabilities of three actions in Gray-CNN.



(c) Probabilities of three actions inthe QST-CNN-LSTM.

Fig. 7 Probabilities of three actions in the three models

Table 3 Recognition accuracy (%) for the UCF sports dataset

Method	Accuracy (%)
3Channel-CNN	58.6
Gray-CNN	77.2
Pyramid CNN [33]	88.1
Dense Trajectories[41]	88.2
CNN-STMH [44]	90.5
Conv-LSTM[7]	92.2
QST-CNN-LSTM	93.2

4.4 Results on UCF11 dataset

In UCF11 dataset, a bounding box of size 110 × 125 is used to extract the key region for the codebook model. An input image of size 44 × 50 is then acquired by using the nearest neighbor algorithm. First, an RGB video clip of size 44 × 50 × 5 (height×width×frames) is used as input for the network. We then apply two quaternion convolutional layers with a kernel of size 7 × 7 × 3 and 5 × 5 × 3 (height×width×channels), with kernel numbers of 4 and 6, respectively. Two temporal convolutional layers of scale 3, and a 2 × 2 and 3 × 3 subsampling operation, are applied following the quaternion convolutional layers. Finally, 180 pure quaternions are calculated in the fully-connected layer as the input for the LSTM. In the experiment, the learning rate is set to 0.1, the batch size is 50, the number of iterations is 10, and the Dropout rate is 0.5.

The confusion matrix of the proposed method is illustrated in Fig. 8, about 5 actions are more than 90% correctly classified on this dataset. However, confusion mainly occur between the similar actions, such as “cycling” and “walking”, “swinging” and “volleyball spiking”. In particularly, the highest accuracy is achieved by the action “diving” (i.e. 100%). Table 4 indicates the performance comparison of different recognition methods. Our method performs better than the best action recognition method based on handcrafted features (i.e. Dense Trajectories). QST-CNN-LSTM also offers better results than the state-of-the-art methods using CNN and LSTM for UCF11 databases. Our method improves results by 0.5% those over results previously reported by Gammulle et al. [7].

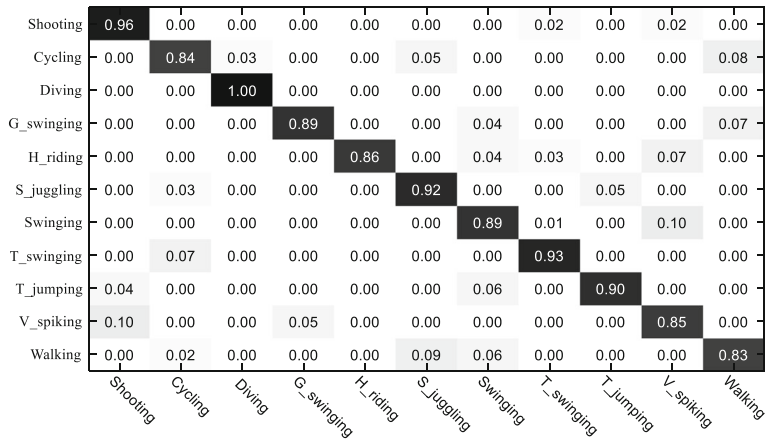


Fig. 8 Confusion matrix for the proposed method for the UCF11 dataset

Table 4 Recognition accuracy (%) for the UCF11 dataset

Method	Accuracy (%)
3Channel-CNN	55.2
Gray-CNN	70.4
Max Pooled LSTM [37]	81.6
Ave Pooled LSTM[37]	82.6
Dense Trajectories[41]	84.2
Conv-LSTM[7]	89.2
QST-CNN-LSTM	89.7

5 Conclusion

We proposed a new deep learning network, QST-CNN-LSTM, for action recognition, integrating QST-CNN and LSTM. The input of the QST-CNN utilizes a quaternion expression for an RGB image, and considers the values of the red, green, and blue channels simultaneously in spatial convolution layers, without losing their spatial relationships. Furthermore, the proposed QST-CNN-LSTM considers the dynamical information of adjacent frames and video clips. Our method achieves higher accuracy on the Weizmann, UCF sports, and UCF11 datasets than that achieved by the state-of-the-art methods.

Our model requires more calculations, because of having to train more parameters than a traditional CNN. The average training times across various experiments on the Weizmann dataset for Gray-CNN and 3Channel-CNN were 20.45 min and 45.20 min respectively. The average training time for QST-CNN-LSTM was 60.04 min. In the future, we will develop our method further to reduce its computational cost and to recognize more complex actions.

Acknowledgments This work was supported by National Natural Science Foundation of China (61602108), Jilin Science and Technology Innovation Developing Scheme (20166016), and the Electric Power Intelligent Robot Collaborative Innovation Group.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Aharon M, Elad M, Bruckstein A (2006) K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311–4322
2. Annane D, Chevrolet JC, Chevret S et al (2014) Two-stream convolutional networks for action recognition in videos. Adv Neural Inf Proces Syst 1(4):568–576
3. Chaudhry R, Ravichandran A, Hager G, et al (2009) Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. Computer Vision and Pattern Recogn, pp. 1932–1939
4. Cheron G, Laptev I, Schmid C (2015) P-CNN: pose-based CNN features for action recognition. pp. 3218–3226
5. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, San Diego, pp 886–893. https://doi.org/10.1109/CVPR.2005.177
6. Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, Darrell T (2017) Long-term recurrent convolutional networks for visual recognition and description. IEEE Transactions on Pattern Analysis & Machine Intelligence 39(4):677–691

7. Gammulle H, Denman S, Sridharan S, Fookes C (2017) Two stream LSTM: a deep fusion framework for human action recognition. In: Winter Conference on Applications of Computer Vision. IEEE, Santa Rosa, pp 177–186. https://doi.org/10.1109/WACV.2017.27
8. Gorelick L, Blank M, Shechtman E et al (2005) Actions as space-time shapes. IEEE Transactions on Pattern Analysis & Machine Intelligence 29(12):2247
9. Guo XL, Yang TT (2016) Dynamic gesture recognition based on kinect depth data. Journal of Northeast Electric Power University 36(2):90–94
10. Gutub AA (2010) Pixel Indicator technique for RGB image steganography. Journal of Emerging Technologies in Web Intelligence 2(1):56–64
11. Gutub A, Al-Juaid N, Khan E (2017) Counting-based secret sharing technique for multimedia applications. Multimedia Tools & Applications 1:1–29
12. Hamilton WR (1969) Elements of quaternions. Vols. I, II. Chelsea Publishing Co, New York
13. Hoai M, Zisserman A (2014) Improving human action recognition using score distribution and ranking. In: Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, pp 3–20. https://doi.org/10.1007/978-3-319-16814-2_1
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
15. Ji S, Yang M, Yu K (2013) 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence 35(1):221–231
16. Kim K, Chalidabhongse TH, Harwood D, Davis L (2005) Real-time foreground-background segmentation using codebook model. Real-Time Imaging 11(3):172–185
17. Kim HJ, Lee JS, Yang HS (2007) Human action recognition using a modified convolutional neural network. In: Advances in Neural Networks - ISNN 2007, 4th International Symposium on Neural Networks. ISNN, Nanjing, pp 715–723. https://doi.org/10.1007/978-3-540-72393-6_85
18. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp 1097–1105.
19. Lan R, Zhou Y (2016) Quaternion-michelson descriptor for color image classification. IEEE Trans Image Process 25(11):5281–5292
20. Lan T, Wang Y, Mori G (2011) Discriminative figure-centric models for joint action localization and recognition. In: International Conference on Computer Vision. IEEE, Barcelona, pp 2003–2010. https://doi.org/10.1109/ICCV.2011.6126472
21. Lan Z, Lin M, Li X, Hauptmann AG, Raj B (2015) Beyond Gaussian Pyramid: multi-skip feature stacking for action recognition. In: Conference on Computer Vision and Pattern Recognition. IEEE, Boston, pp 204–212. https://doi.org/10.1109/CVPR.2015.7298616
22. Lan ZZ, Yu S-I, Yao D, Lin M, Raj B, Hauptmann AG (2016) The Best of BothWorlds: combining data-independent and data-driven approaches for action recognition. In: Conference on Computer Vision and Pattern Recognition Workshops. IEEE, Las Vegas, pp 1196–1205. https://doi.org/10.1109/CVPRW.2016.152
23. Laptev I, Lindeberg T (2005) On space-time interest points. Int J Comput Vis 64(2–3):107–123. https://doi.org/10.1007/s11263-005-1838-7
24. Le QV, Zou WY, Yeung SY, Ng AY (2011) Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: The 24th Conference on Computer Vision and Pattern Recognition. IEEE, Colorado Springs, pp 3361–3368. https://doi.org/10.1109/CVPR.2011.5995496
25. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos. In: Computer vision and Pattern Recognition. IEEE, Miami, pp 1996–2003. https://doi.org/10.1109/CVPRW.2009.5206744
26. Liu Z, Zhang C, Tian Y (2016) 3d-based deep convolutional neural network for action recognition with depth sequences. Image Vis Comput 55:93–100
27. Ng JYH, Hausknecht MJ, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: deep networks for video classification. In: Computer vision and Pattern Recognition. IEEE, Boston, pp 4694–4702. https://doi.org/10.1109/CVPR.2015.7299101
28. Norah A, Basem A, Adnan G (2017) Applicable light-weight cryptography to secure medical data in IoT systems. Journal of Research in Engineering and Appl Sci 2(2):50–58
29. Palm RB (2012) Prediction as a candidate for learning deep hierarchical models of data. Technical University of Denmark, Lyngby, pp 1–87
30. Park E, Han X, Berg TL, Berg AC (2016) Combining multiple sources of knowledge in deep CNNs for action recognition. In: Winter Conference on Application of Computer Vision. IEEE, Lake Placid, pp 1–8. https://doi.org/10.1109/WACV.2016.7477589
31. Peng X, Qiao Y, Peng Q (2014) Motion boundary based sampling and 3D co-occurrence descriptors for action recognition. Image Vis Comput 32(9):616–628
32. Peng X, Zou C, Qiao Y, Peng Q (2014) Action recognition with stacked fisher vectors. In: Computer Vision - 13th european conference. ECCV, Zurich, pp 581–595. https://doi.org/10.1007/978-3-319-10602-1_38

33. Ravanbakhsh M, Mousavi H, Rastegari M, Murino V, Davis LS (2015) Action recognition with image based CNN features, CoRR abs/1512.03980. https://dblp.org/rec/bib/journals/corr/RavanbakhshMRMD15. Accessed 07 Jun 2017
34. Rodriguez MD, Ahmed J, Shah M (2008) Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In: Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, Anchorage. https://doi.org/10.1109/CVPR.2008.4587727
35. Salih AAA, Youssef C (2016) Spatiotemporal representation of 3D skeleton joints-based action recognition using modified spherical harmonics. Pattern Recogn Lett 83:32–41
36. Sapienza M, Cuzzolin F, Torr PHS (2014) Feature sampling and partitioning for visual vocabulary generation on large action classification datasets. CoRR abs/1405.7545. http://arxiv.org/abs/1405.7545. Accessed 07 Jun 2017
37. Sharma S, Kiros R, Salakhutdinov R (2015) Action recognition using visual attention. CoRR abs/1511.04119. http://arxiv.org/abs/1511.04119. Accessed 07 Jun 2017
38. Sivic J, Zisserman A (2003) Video Google: a text retrieval approach to object matching in videos. In: 9th International Conference on Computer Vision. IEEE, Nice, p 1470–1477. https://doi.org/10.1109/ICCV.2003.1238663
39. Tian YY, Tan QC (2016) Sub-pixel edge localization algorithm for filtering noise analysis. Journal of Northeast Electric Power University 5:56–60
40. Wang H, Schmid C (2014) Action recognition with improved trajectories. In: International Conference on Computer Vision. IEEE, Sydney, pp 3551–3558. https://doi.org/10.1109/ICCV.2013.441
41. Wang H, Klaser A, Schmid C, Schmid C, Liu C-L (2011) Action recognition by dense trajectories. In: The 24th Conference on Computer Vision and Pattern Recognition. IEEE, Colorado Springs, pp 3169–3176. https://doi.org/10.1109/CVPR.2011.5995407
42. Wang JW, Le NT, Lee JS et al (2016) Color face image enhancement using adaptive singular value decomposition in fourier domain for face recognition. Pattern Recogn 57(C):31–49
43. Wang L, Ge L, Li R et al (2017) Three-stream CNNs for action recognition[J]. Pattern Recogn Lett 92(C):33–40
44. Weinzaepfel P, Harchaoui Z, Schmid C (2015) Learning to track for spatio-temporal action localization. In: International Conference on Computer Vision. IEEE, Santiago, pp 3164–3172. https://doi.org/10.1109/ICCV.2015.362
45. Wu K, Li Gui J, Han GL (2017) Color image detail enhancement based on quaternion guided filter. Journal of Computer-Aided Design & Computer Graphics 29(3):419–427
46. Yang X, Zhang C, Tian Y (2012) Recognizing actions using depth motion maps-based histograms of oriented gradients. In: MM'12 Proceedings of the 20th ACM international conference on Multimedia. Nara, pp 1057–1060. https://doi.org/10.1145/2393347.2396382
47. Zeng R, Wu J, Shao Z, Chen Y, Chen B, Senhadji L, Shu H (2016) Color image classification via quaternion principal component analysis network. Neurocomputing 216:416–428
48. Zou C, Kou KI, Wang Y (2016) Quaternion collaborative and sparse representation with application to color face recognition. IEEE Trans Image Process 25(7):3287–3302

Bo Meng received her M.S. and Ph.D degrees in Changchun Institute of Fine Mechanics and Physics, Chinese Academy of Sciences China in 2005 and 2008. She is an associate professor of Northeast Electric Power University since 2011, and did researching work at Laboratory of Computational Sensing Robotics of Johns Hopkins University, USA as a visiting scholar in 2014. Her research interests include computer vision, object recognition, and object tracking.

Xuejun Liu received her bachelor's degree in Binzhou University in 2015 and now she is a graduate student of computer science and technology in Northeast Electric Power University. Her research directions include computer vision and action recognition.

Xiaolin Wang received her bachelor's degree in Hebei Finance University in 2016 and now she is a graduate student of computer science and technology in Northeast Electric Power University. Her research directions include computer vision and action recognition.