Applied Intelligence (2022) 52:12771–12787
https://doi.org/10.1007/s10489-021-03068-w

Hybrid handcrafted and learned feature framework for human action recognition

Chaolong Zhang[1,2] · Yuanping Xu[2] · Zhijie Xu[1] · Jian Huang[2] · Jun Lu[2]

Accepted: 30 November 2021 / Published online: 12 February 2022

Abstract

Recognising human actions in video is a challenging task in real-world. Dense trajectory (DT) offers accurate recording of motions over time that is rich in dynamic information. However, DT models lack the mechanism to distinguish dominant motions from secondary ones over separable frequency bands and directions. By contrast, deep learning-based methods are promising over the challenge though still suffering from limited capacity in handling complex temporal information, not mentioning huge datasets needed to guide the training. To take the advantage of semantical meaningful and "handcrafted" video features through feature engineering, this study integrates the discrete wavelet transform (DWT) technique into the DT model for gaining more descriptive human action features. Through exploring the pre-trained dual-stream CNN-RNN models, learned features can be integrated with the handcrafted ones to satisfy stringent analytical requirements within the spatial-temporal domain. This hybrid feature framework generates efficient Fisher Vectors through a novel Bag of Temporal Features scheme and is capable of encoding video events whilst speeding up action recognition for real-world applications. Evaluation of the design has shown superior recognition performance over existing benchmark systems. It has also demonstrated promising applicability and extensibility for solving challenging real-world human action recognition problems.

Keywords Action recognition · Dense trajectories · Bag-of-temporal features · Visual stream · Motion stream

1 Introduction

Recognising and understanding in the complex visual world is a relatively easy job for humans but a much more challenging task for computers. Computer vision has been a long-lasting research hotspot for over half-a-century with prominent discoveries and breakthroughs in every decade, namely a few, pictorial and geometrical representation in the 70s, quantitative image and scene analysis in the 80s, recognition in the 90s, feature engineering at the turn of the millennium, and deep-learning in the 2010s. Human behaviour analysis is one of the most intriguing research areas in computer vision due to its wide range of applications in intelligent video surveillance, abnormal behaviour detection, novel human computer interface (HCI) design, and even game and entertainment [28]. However, it remains a challenging task even now due to semantic implicit and ambiguous definition of video events, for example, the classification and categorization of individual and crowd motions, never mention the inherent signal complexities from recorded or streamed videos ill-affected by target occlusion and variation of illumination conditions [28].

Intuitively, an action is considered as a human agent performing a sequence of basic (or atomic) movements. Historically, there are two main research strategies for human action recognition: a) using "handcrafted" features and labelled patterns of them for representing and identifying action types; and 2) using semi or fully automatically "learned" features in an end-to-end manner for classifying behaviours. The prior follows a bottom-up strategy, which consists of three main phases: foreground detection (e.g., using Gaussian Mixture Model); feature extraction and representation (e.g., Scale Invariant Feature Transform [18], Harris detector [13], and dense sampling [40]); and action classification (e.g., Support Vector Machine and Artificial Neural Network). One of the popular process pipelines employs the dense trajectories (DT) model [40] and its varied versions

✉ Yuanping Xu
ypxu@cuit.edu.cn

[1] School of Computing and Engineering, University of Huddersfield, HD1 3DH Huddersfield, UK

[2] School of Software Engineering, Chengdu University of Information Technology, Chengdu, China

for the final phase, such as the improved dense trajectory (iDT) [41], stacked fisher vector (SFV) [24], and the discriminative action vector model (DA-VLAD) [23]. Since it is birth in 2012, iDT had become the baseline for performance evaluation in video event analysis. It remains a widely adopted benchmark even in the deep learning era. However, dense trajectory models lack the mechanism to distinguish dominant motions from secondary ones for differentiating human actions over separable frequency bands and directions. This study explores the integration of wavelet techniques into the dense trajectory domain for gaining more descriptive human action patterns, so as to better harness the advantages from the semantically more representative handcrafted video features.

Machine learning, especially the recent trend of deep learning (DL), supports direct feature abstraction (modelling) and pattern recognition (classification), which has become a mainstream pipeline for image understanding due to its computational brute-force approach and the robustness for certain application tasks such as image classification [3] and object detection [19, 25]. Comparing to handcrafted features that are heavily dependent on complex feature design such as interest points, the ground-breaking CNN (Convolutional Neural Network) model avoided the laborious feature crafting step, hence initiating a paradigm-shift from an "engineering" one to an "architectural" one. Many DL frameworks have since been piloted producing varied level of "performance gain" in different signal spectrum, from spatial, frequency to temporal. In the human action recognition area, Karpathy et al. presented an extensive empirical evaluation of CNNs on video-based action classification [9], which shows a strong potential of CNN on action recognition. A series of CNN-based models then followed, including Two-stream CNN [29], C3D (3D convolutional networks) [35], semantic adaptation model (SAM) [52], and Parallel Pair Discriminant Correlation Analysis (PPDCA) [48]. Two-stream models show superior performance by joining the video features from spatial and temporal domains. Various forms of two-stream architectures had been developed, including the hidden two-stream [56], two-stream with LSTM [4], spatiotemporal pyramid network [44], and the two-stream feature fusion CNN (TSFFCNN) [47], namely a few. However, one question remains on how to bridge the semantic gap between man-made features, which often carrying distinctive "meanings", and the "automated" latent ones embedded in the ever-sprawling webs of deeper layers.

To tackle the current shortcomings, this study proposes a hybrid handcrafted and deep learning feature framework to improve the performance of real-time action recognition through harnessing the advantages of engineered features and the learned ones. This framework deploys a Direct Wavelet Transform (DWT) on input frames to ensure effective video event separation and representation. A DT extraction operation is then engaged to track feature points and to form the so-called trajectory feature vectors consisting of shapes, histogram of oriented gradients, histogram of optical flows, and motion boundary histograms. A pretrained dual-stream CNN-RNN (Convolutional Neural Network and Recurrent Neural Network) feature extractor is devised through adopting the C3D [35] and the VGG [30] networks. The parameters of the dual-stream network are inherited from previous studies [29], including the training on the ImageNet ILSVRC-2014 classification dataset [30] and the fine-tuning on the chosen human action-specific datasets. The proposed hybrid framework then integrates the "learned" spatial-temporal features with the "handcrafted" ones by forming fisher vectors before it is classified based on the devised Bag-of-Temporal-Features (BoTF) event representation scheme.

The rest of this paper is organized as follows. Section 2 presents a brief review over the concepts and related works in the field. Based on the studied literatures, the rationales for the proposed approach are justified. In Section 3, an overview of the devised hybrid framework for human action recognition is elaborated through highlighting the implementation details of the DWT for extracting dense trajectories of motions and the integration strategy for the DL models. Human action taxonomy and representation techniques adopted in this research are presented in Section 4 that includes an innovative fisher vector encoding scheme and a novel BoTF representation method for enabling a Support Vector Machine (SVM) classifier for sematic action recognition. Section 5 illustrates experiments carried out in the research and evaluations against relevant benchmarks. Finally, Section 6 concludes the research with anticipated future works.

2 Concepts and related works

According to the human behaviour complexity and their semantic definition, human actions can be classified into five categories: gestures, individual actions, human-object interactions, human-human interactions, and group activities [28]. A gesture is defined as a basic or atomic movement of the human body parts that carries some meaning, e.g., 'head shaking', 'hand waving', and 'facial expression'. An individual action is a type of activity that performed by a single person, where 'walking', 'running' and 'jumping' are cases of it. Interaction is performed by at least two enactors that can be divided into human-human interaction, e.g., handshaking, fist-fighting, or wrestling between two persons; and human-object interaction, e.g., a person using a phone and a person accessing an ATM. Group activity is also named as crowd behaviour that is the most complex type of activity that may combines gestures, actions, and interactions, for example, cheerleading and Marathons. This study concentrates on application scenarios involving individual actions and is extended to some of the human-human and human-object interactions.

2.1 Human action recognition using "handcrafted" features

In the so-called "handcrafted" feature approach, Gaussian Mixture Model (GMM) is the mainstream algorithm for foreground detection. GMM assumes that the background is more stable than the foreground, though may lead to the loss of high-speed targets. To solve this problem, Droogenbroeck et al. presented the ViBe model for background subtraction [36]. Wu et al. proposed a Scale Invariant Feature Transform (SIFT) based model to extract feature points [45]. SIFT can robustly extracts features from images because of its invariance to uniform scaling, orientation and illumination changes. However, SIFT fails to handle three-dimensional (3D) data volumes (e.g., videos). Liu et al. extended SIFT to 3D space that can extract interest points from 3D space-time video volumes efficiently [18], and then the Latent Dirichlet Allocation (LDA) model was integrated for human action classification. Laptev and Lindeberg introduced space-time interest points by extending the Harris detector with significantly improved detection rate [13]. Sipiran et al. improved the Harris operator to Harris 3D model that can extract interest points from 3D data volumes effectively [31]. Generally, the space-time feature point approaches have shown sound effectiveness. Wang et al. presented a Spatial Temporal Volume (STV) based model for action detection from CCTV (closed-circuit television) clips [42]. STV-based approaches are suitable for recognizing simple gestures and actions (e.g., hand waving and walking), but it falls short to capture complex actions (e.g., "TaiChi" action in the UCF 101 dataset), which is especially true when recognising multiple person-based activities.

Trajectories based methods show reasonable results on several datasets [51]. Messing et al. developed a Harris 3D and Kanade-Lucas-Tomasi (KLT) based tracking model to track feature points and obtain trajectory features from videos [22]. Sun et al. developed a SIFT based tracker to obtain trajectories [8]. Later, Sun et al. combined the two trackers to increase the density of trajectories [33]. However, both KLT and SIFT trackers are still insufficient to handle the frame boundaries and describe complex motion patterns. Thus, Wang et al. presented a dense trajectories (DT) framework to tackle this problem [40]. DT model densely samples feature points on each spatial scale, and then tracks the points in the next frames with a preset length l. The trajectories $(P_1, P_2, ..., P_l)$ are obtained when the number of tracked frames is completed, where P_i indicates a feature point in i-th frame. Aligned with the trajectories, four features are extracted, including trajectory shapes (TS), histogram of oriented gradients (HOG), histogram of optical flows (HOF), and motion boundary histogram (MBH). After that, the Bag-of-features (BoF) concept is applied for feature assembly. Dense trajectories model is more robust to handle complex motion patterns when compared with KLT and SIFT.

Since its appearance, the DT model has been gaining popularity and being tested on various action datasets with significant improvements over the state-of-the-art at that time. It had drawn wide attention and optimism since [6, 24, 41]. Wang et al. further improved their works (named iDT) by investigating Speeded Up Robust Features descriptor (SURF) and fisher vectors (FV) [41]. Jiang et al. developed an action prediction method based on dense trajectories and dynamic image models [6], which is capable of predicting evolutional trends of actions in videos. Peng et al. proposed Stacked Fisher Vectors (SFV) with multi-layer nested fisher vector encoding for human action recognition [24]. SFV can refine the representation and abstract semantic information in a hierarchical way, which has improved capacity for encoding combinatory features hence improving classification accuracy. However, the BoF in DT encodes the four type features as an unordered set, and the spatial and temporal information are largely ignored. In 2013, Bolovinou et al. presented the Bag of Spatio-Visual Words (BoSVW) to encode ordered spatial information for scene classification [1]. Later, Zhao et al. further improved this model by combining multiscale features, and it gained better performance on scene classification [55]. BoSVW significantly improved the BoF model through integrating spatial context. Inspired by these achievements, this study explores a so-called bag-of-temporal-features (BoTF) technique to encode temporal information, i.e., it can encapsulate the ordered motion information (temporal features) of an action in a video clip.

In terms of action classification, SVM is a dominant model that has shown superior performance over others on most classification tasks, such as image classification and object detection [2]. In recent years, Neural Network (NN) and LDA models had emerged as effective methods for classification applications [18, 38].

2.2 Human action recognition by deep learning models

Deep learning models extract features automatically from the input data. Of this "unsupervised" style, it has gained tremendous popularity in many applications. For example, image classification tasks have experienced almost a complete overhaul through varied forms of CNN implementations. Object detection and facial recognition have also achieved encouraging results [7, 46].

Recently, deep learning based human action recognition has seen major breakthroughs, including the dual-stream CNN model proposed by Simonyan et al. [29], and the improved dual-stream models, e.g., hidden two-stream [56], two-stream with LSTM models [4], and the two-in-one stream model proposed by Zhao in 2019 [53]. To tackle the disadvantage of lacking of time-scale diversity in the temporal domain, Wan et al. developed a dual-stream convolutional network with the long-short-term spatiotemporal features (LSF CNN) [39] which indicates a promising direction for consistently handling motion features in both spatial and temporal domains. Another line of interesting work focused on tackling three-dimensional (3D) data (e.g., videos) using 3D CNN. For instance, Ji et al. developed a 3D CNN model for surveillance video analysis [5]. Tran et al. presented a C3D (Convolutional 3D) feature learning model with $3 \times 3 \times 3$ convolutional kernels in all layers [35]. CNN, especially 3D CNN, shows sound performance in general on visual classification tasks. However, this design only tracks a short time period for the temporal features in video clips, e.g., 16 frames, that leads to difficulty when dealing with "longer" event sequences. Another significant drawback of current CNN implementation is its limitation in dealing with sequential temporal information such as plots in movies. In 2017, Li et al. introduced a Recurrent Neural Network (RNN) based Long-short Term Memory (LSTM) model for handing spatial-temporal features [16]. Shortly after, Majd et al. presented a correlational convolutional LSTM (C^2 LSTM) to handle both the spatial and motion structure of surveillance video data [21]. Wang et al. presented a so-call trajectory-pooled deep-convolutional descriptor (TDD) that embeds the features from both handcrafted and deep-learning models [43]. Motivated by TDD, Lu et al. developed a multi-scale trajectory-pooled 3D convolutional descriptor (MTC3D) by combining dense trajectories and 3D CNN [20]. TDD and MTC3D are capable of automated learning of temporal features from motion trajectories. However, they have fallen short to capture long-term temporal information. To alleviate this major problem, this study proposed an innovative technique to integrate time information with motion features extracted from trajectories, so that "longer" temporal events can be annotated explicitly (refer to BoTF event representation in Section 4.2).

3 Motion feature extraction

The process pipeline of the proposed human action recognition model is shown in Fig. 1, in which an input video is pre-processed for feature point extraction and tracked by the DWT enabled DT model. The outputs are a series of low-level handcrafted features (see Fig. 2) describing the trajectory patterns inherited from the STV data. Then, the STV data feeds into the pre-trained dual-stream CNN-RNN and 3D CNN model for extracting the learned spatial-temporal features. Both handcrafted and learned features are then encoded into Fisher Vectors annotated by the proposed BoTF representation scheme (detailed in Section 4). Finally, all video features are fused into a holistic video event representation scheme. It will then be classified by a SVM classifier for action recognition. Sub-sections below explain the relevant operations in detail.

3.1 DWT-based decomposition

Traditional DT-based approaches extract feature points and then track them in video frames, which lacks detail and interpretable information on separable frequency and movement orientation. Wavelet transform has the ability of recording coarse-to-fine presentation of spatial features. It has been demonstrated that DWT models can not only dissecting an image in the form of multi-resolution representations but also extracting textural features representing motion characteristics, hence contributing to semantic feature representation such as the Bag-of-Words models [54]. Inspired by the pilot work, the proposed technique decomposes video frames into different frequencies and orientations of multiple scales through applying the DWT filter as shown in Fig. 2. The single level 2D DWT is applied to decompose a video frame into A, H, V and D parts as shown in Fig. 3, where A is the approximation coefficients and H, V and D donate detailed coefficients along horizontal, vertical and diagonal orientations respectively. Compared with the original video frame, these four components are of lower total size, and A contains information of overall context. In contrast, H, V and D possess dominant movement information along varied orientations. Hence this approach enables a more effective feature extraction and tracking model.

3.2 Dense trajectory formation

This study samples feature points densely on a grid of 5×5 for the input frames. In this process, the first spatial scale data (I) is the input frame itself. Then its spatial scale increases by a factor of $1/\sqrt{2}$. To reduce the amount of trivial and redundant feature points in homogeneous areas, a threshold T is deployed on the eigenvalues for each scale as shown in the following equation:

$$T = k \times \underset{i \in I}{maxmin}(\lambda_i^1, \lambda_i^2) \quad (1)$$

where $(\lambda_i^1, \lambda_i^2)$ are the eigenvalues of i-th point in the spatial scale data I and its corresponding DWT coefficients. The k is taken as 0.001 for A, H and V of the original spatial scale data, while k is set as 0.01 for D. Dense sampling across

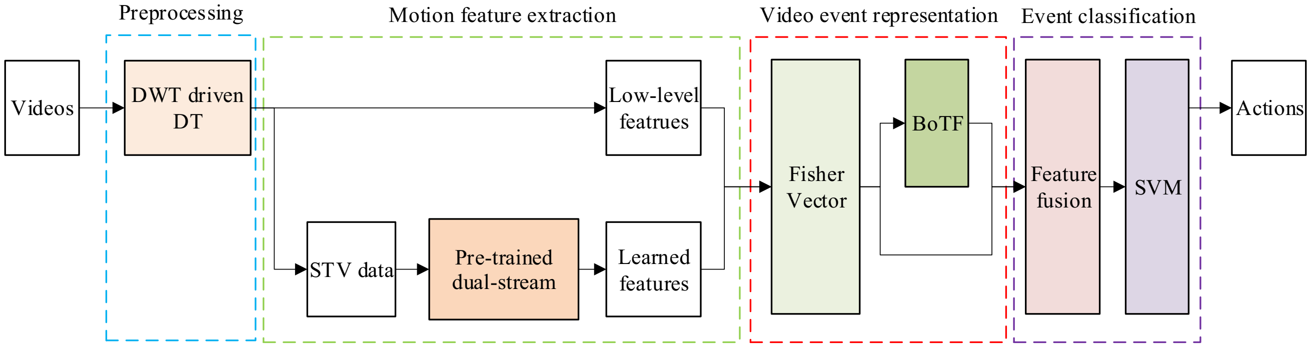


Fig. 1 The process pipeline of the human action recognition model. It contains four stages. The raw pixel data are pre-processed by DWT and DT. After getting the low-level and learned features by the motion feature exactions from training videos, Fisher Vector and BoTF schemes are applied to generate the codebook. At the end of the pipeline, SVM is used for action recognition

all spatial scales ensures the comprehensiveness of feature points extracted and their motion potentials. For example, Fig. 4 demonstrates the feature points extracted from the original (first) spatial scale, while Fig. 5 illustrates the feature points extracted from the downward scales.

Feature points from continuous input frames are then batch processed and tracked on each spatial scale respectively before median filtering are performed on the dense optical flow fields m_t. The feature point tracking strategy is shown as following:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * m_t) \tag{2}$$

where P_{t+1} is a tracked point in the consecutive frames, M is a median filtering kernel with the size of 3×3, and (x_t, y_t) indicates a feature point in t-th frame, and m_t is the dense optical flow.

The length of a typical action tracked is set at 15 frames (roughly two thirds of a second) based on human behavioural studies [40]. Once a tracked action is completed, a trajectory

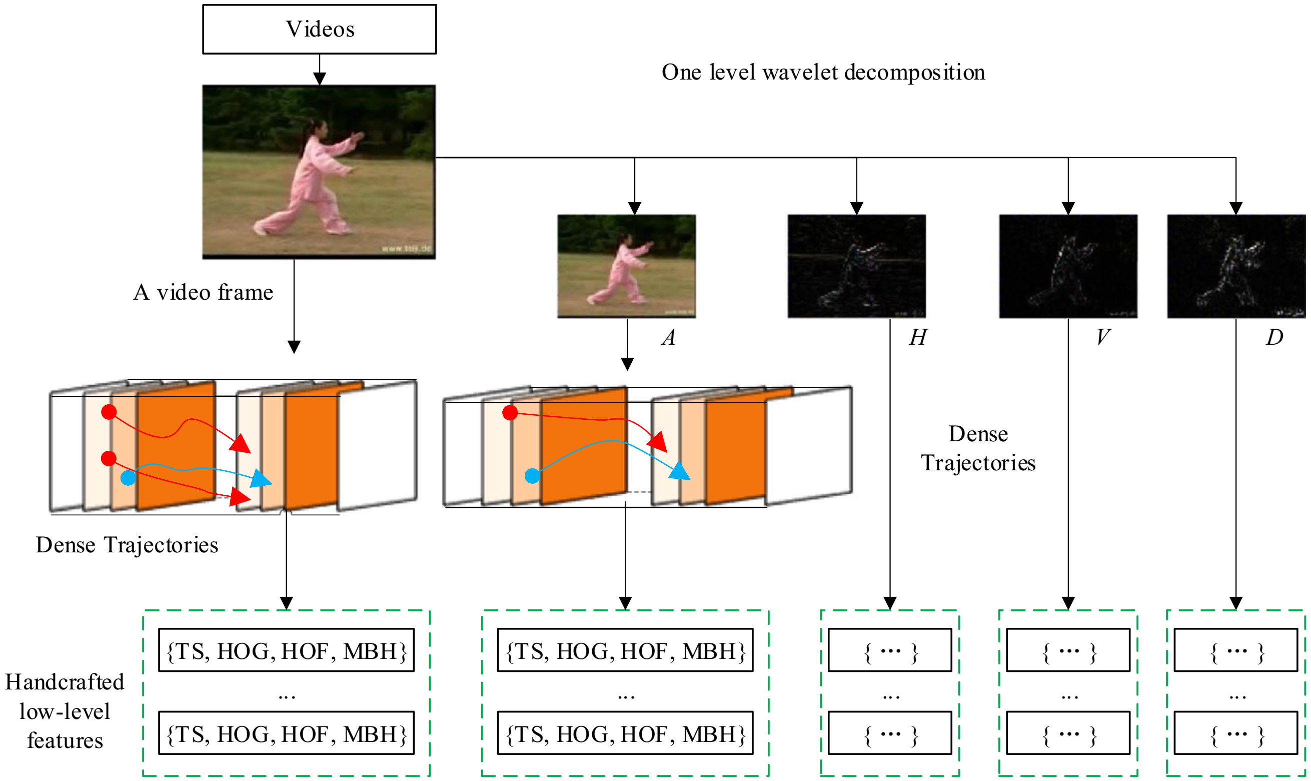
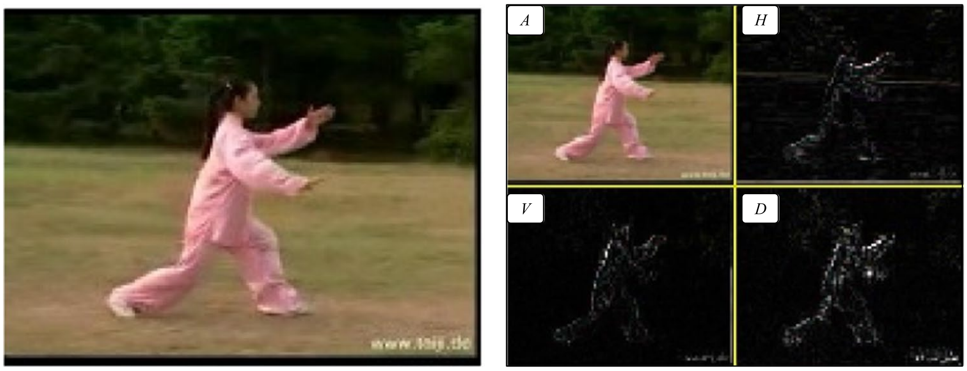


Fig. 2 The processing steps of DWT driven DT based feature extractor. This study decomposes the original video frames into four coefficients. Along with the original frame, DT method is applied to generate the trajectories (red and blue curves) and the low-level features



(a) A original video frame (b) The DWT decomposition result

Fig. 3 A demonstration of DWT pre-processing for a video frame coming from UCF 101 action dataset. (a) illustrates a video frame from a "TaiChi" action video clip, while (b) shows the corresponding DWT coefficients

will be obtained in the form of $(P_t, P_{t+1}, P_{t+2}, ..., P_{t+14})$. For storing feature trajectories, this study has devised a STV (Spatial Temporal Volume) structure for encapsulating motions denoted by tracked features from all 15 frames as shown in Fig. 6. The design ensures a compact and comprehensive representation of motion and context information inherited from a video event (human action) under study.

3.3 Low-level feature extraction

Four handcrafted motion and contextual features:

Trajectory Shapes (TS) are denoted by a vector $(\Delta P_t, \Delta P_{t+1}, \Delta P_{t+2}, ..., \Delta P_{t+14})$ based on a trajectory $(P_t, P_{t+1}, P_{t+2}, ..., P_{t+14})$, in which

$$\Delta P_t = (\Delta x_t, \Delta y_t) = P_{t+1} - P_t \tag{3}$$

TS records the normalized derivative of the trajectory tendency that can be calculated as the following:

$$TS = \frac{(\Delta P_t, \Delta P_{t+1}, \Delta P_{t+2}, ..., \Delta P_{t+14})}{\sum_{i=t}^{t+14} ||\Delta P_i||} \tag{4}$$

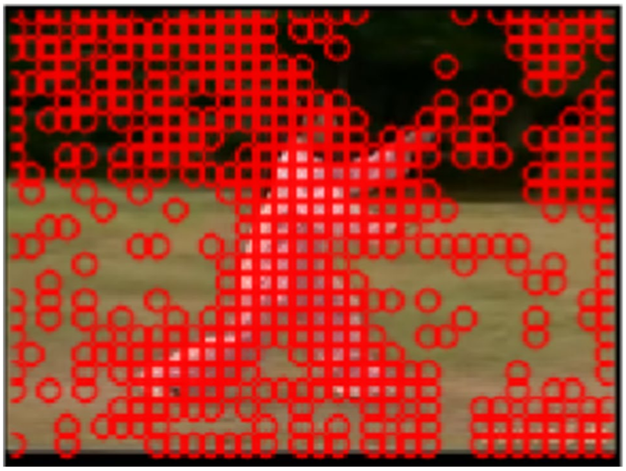


Fig. 4 Feature points extracted from original spatial scale

where $||x||$ indicates the L2-norm. *TS* calculation is rooted into the tracked point coordinates, so it reflects the shape information of a trajectory representing movements at each spatial scale and orientation. As the length of a trajectory is fixed at 15 frames and each point contains 2-dimensional coordinates, so a single *TS* descriptor is a 30-component vector.

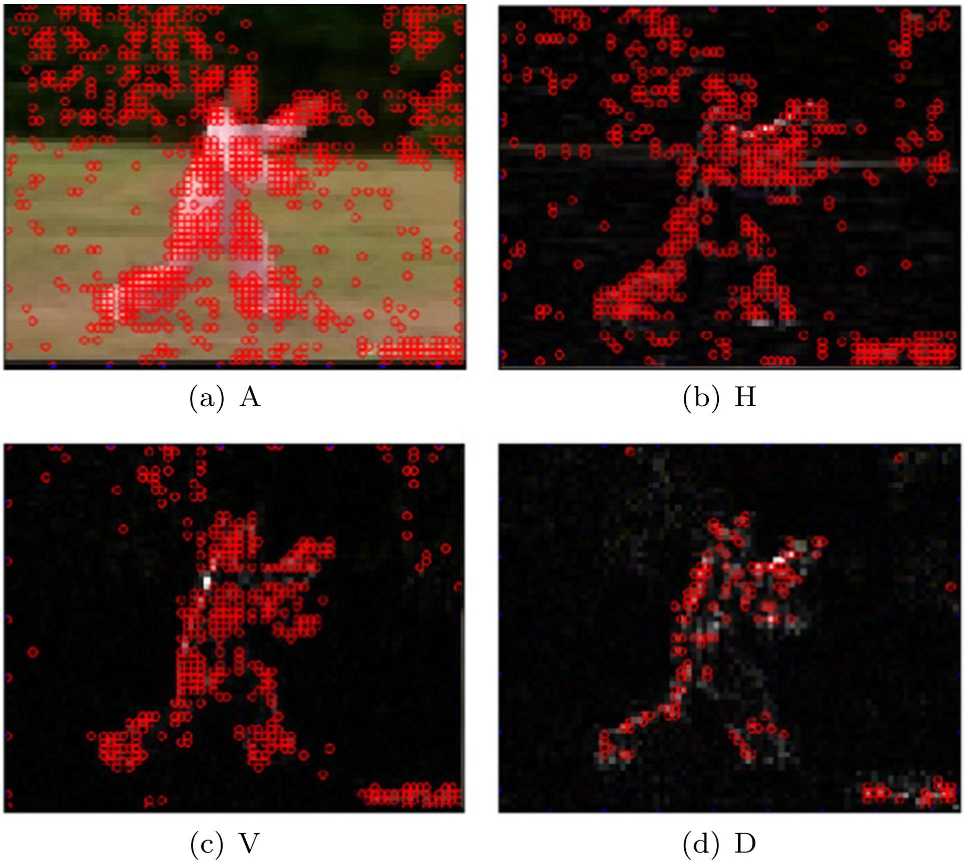
Histogram of Oriented Gradients (HOG) formulates motion and appearance information from a STV block. This study computes HOG [14] based on the encapsulated STV data holding neighbourhood feature points. A complete STV block is subdivided into a (n_x, n_y, n_t) grid of cuboids, where $n_x = n_y = 2$, and $n_t = 3$, and the green cell in Fig. 6 is one of the subdivided cuboid, which is inspired by the work reported earlier [40]. For each cuboid, this study computes HOG by applying the algorithm of Laptev [14], and sets the number of bins to 8, so the final HOG descriptor will output a 96-component vector.

Histogram of Optical Flow (HOF) descriptor is similar to HOG except that the input data is populated by the extracted dense optical flow. Optical flow is a significant feature for video processing. It tracks the motion information between two sequential frames, such that the HOF can encode the movements efficiently. This study computes HOF of each cuboid and the number of bins has been increased to 9 to accommodate the zero bin often occur in practice. Unlike Wang's work [40] that inherits the optical flow algorithm from Farnebäck, this study applies the so-called SimpleFlow [34] because of its compactness and efficiency. The HOF descriptor outputs a 108-component vector.

Motion Boundary Histogram (MBH) is proposed here to correct the camera motion that often occurs in real life videos. In the MBH, local constant camera motions are removed while preserving human motions through computing derivatives of optical flows, as show in the following:

$$\begin{cases} {w_x}' = dw_t|(x_t, y_t)dx = \frac{\partial w_t|(x_t,y_t)}{\partial x} \\ {w_y}' = dw_t|(x_t, y_t)dy = \frac{\partial w_t|(x_t,y_t)}{\partial y} \end{cases} \tag{5}$$

Fig. 5 Feature points extracted from DWT coefficients


(a) A

(b) H

(c) V

(d) D

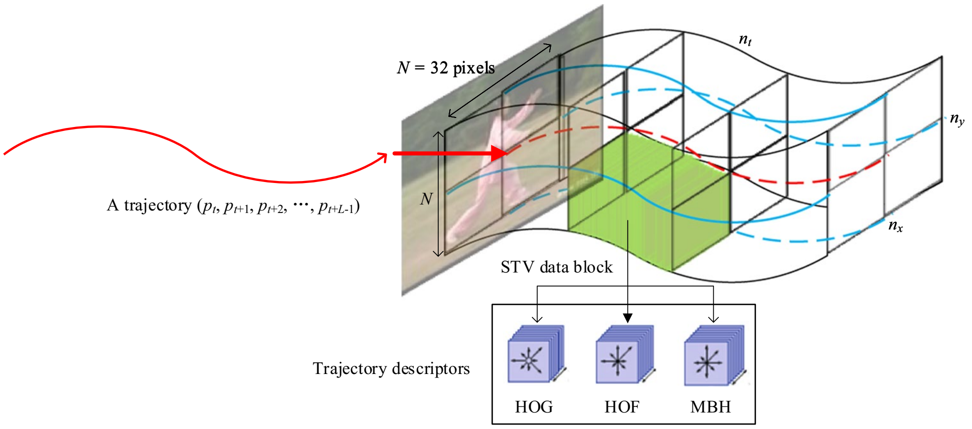
where w' is the motion boundary defined by differential coefficient of dense optical flow. This study sets an 8-bin histogram along x- (MBH_x) and y-axis (MBH_y) for each cuboid. The MBH descriptor generates a 192-component vector.

All those low-level feature descriptors are normalized by L2-norm and combined into 426-component feature representation vector v, as shown in the following:

$$v = (\alpha TS, \beta HOG, \gamma HOF, \delta MBH) \tag{6}$$

where α , β , γ and δ indicate the weights of each feature respectively. This study treats all features with equal weight initially, and future work will see these weights being "learned" based on application scenarios dynamically. Compared with classic DT approaches, this study has validated a more comprehensive and robust feature vector for extracting important trajectory features from the multiple spatial scales.

Fig. 6 The encapsulated STV block for storing feature trajectories. The left red curve is a trajectory that is constructed by 15 tracked points



3.4 Pre-trained feature adaptation

Deep feature methods have shown outstanding performance on image classification and video analysis. However, training a CNN model is very time-consuming and requires a great deal of labelled data, which limits the applications of CNN models. This study considers the pre-trained CNN models as general feature extractors. A CNN architecture can be depicted as the follows:

$$\begin{cases} F = f(wI, b) \\ G = g(w'F, b') \end{cases} \tag{7}$$

where w and w' are learnable weights belonging to CNN kernels and the fully-connected layers respectively, b and b' are learnable bias, I indicates an image or a video frame, $f(*)$ is a learnable function which presents the CNN layers of a deep learning architecture, while $g(*)$ indicates the fully-connected layers. According to the literatures [50], $f(*)$ is more generic and it extracts features such as interest points, lines and edges in different CNN layers, so it is reasonable to apply the pre-trained CNN models to extract features. In practice, this study extracts the vector values from the fully-connected layer of the networks and removes the rest layers since only the specified features from CNN models are of interest. In this study, only the first (fc_1) and second (fc_2) fully-connected layers of G in (7) are kept. Here, it is worth noting that various pre-trained networks and fully connected layers can be integrated, which will be evaluated in Section 5.4

From a given video clip, the trajectory encapsulated in the STV block can be formulized as $V = \{I_i | i \in [1, L]\}$. This study firstly extracts the learned image feature h_i from each frame I_i by the pre-trained CNN model, here f_i is a 2049-dimensional feature vector. A series of image feature vectors $H = \{h_i | i \in [1, L]\}$ of the trajectory can be obtained, and then the feature fusion method is applied by averaging the series of feature vectors, which outputs a trajectory-level feature p, where:

$$p = \frac{1}{N} \sum_{i=1}^{N} h_i, N = L \tag{8}$$

In principle, any type of CNN architecture can be adopted for action feature extraction, such as AlexNet [11] and VGG networks [30]. Both of them were pre-trained on the ImageNet ILSVRC-2014 classification dataset. This study explores a dual-stream CNN architecture due to its distinctive capacity of encoding visual (RGB) and motion (optical flow) features simultaneously in the spatial and temporal domains. The dual-stream CNN model has achieved reasonable results on human action recognition at the accuracy of 88.0% and 59.4% on UCF 101 and HMDB51 respectively [29], and the performance achieves the benchmark level of the improved DT method in pre-DL era. However, the dual-stream CNN model neglects the intrinsic differences between temporal and spatial domains. To alleviate this shortcoming, the devised framework incorporates the strengths of both the 3D CNN in spatial domain and the RNN for handling the temporal features, the whole network design is shown in Fig. 7.

The visual (RGB) stream is comprised of by two components: the 3D CNN-based "appearance" feature extractor and the RNN-based sequential descriptor. In practice, this study applies the C3D model as the CNN component. C3D uses 3D convolution and 3D pooling operations on each layer. This study uses $3 \times 3 \times 3$ convolution kernel for convolution layers and all pooling layers are max pooling with kernel size $2 \times 2 \times 2$. With this configuration, C3D is trained on 15 consecutive frames (STV) with the input size of 3 (channel) × 15 (frames) × 112 (pixel) × 112 (pixel) and outputs 2049 units in the last fully-connected layer, which is followed by a RNN structure for sequential modelling. The RNN units support the sequential learning, i.e., learning connections between inputs and the corresponding previous states continuously, which is ideal for extracting temporal information in videos. A LSTM model has been applied in this design instead of the traditional RNN model for its unique ability in remembering "states" over a long period of time by using the "forget" mechanism. The devised RNN structure has two LSTM layers, and each of them has 1024 hidden states, so the RNN component outputs a 1024-component feature vector.

The motion (optical flow) stream is also constructed by the CNN and RNN components. Different from the visual stream, the motion stream mainly extracts temporal event features from the successive flow fields. This study adapts the VGG-16 network as the CNN component. With this configuration, VGG-16 is trained on a stacked optical flow computed from the STV block, so the input size is 2 (channel) × 15 (frames) × 112 (pixel) × 112 (pixel) and the output is a 2049-component vector in the last fully connected layer that is followed by a RNN for sequential modelling.

4 Video event representation

4.1 Fisher vector encoding

Let $X = \{x_i, i \in [1, T]\}$ be the series of low-level features extracted and formulated as shown earlier. Noted that x_i can be various of features such as a single feature e.g., *TS*, *HOG*, *HOF*, *MBH* or a learned feature, or a hybrid one through combination. Fisher vector assumes the generation process

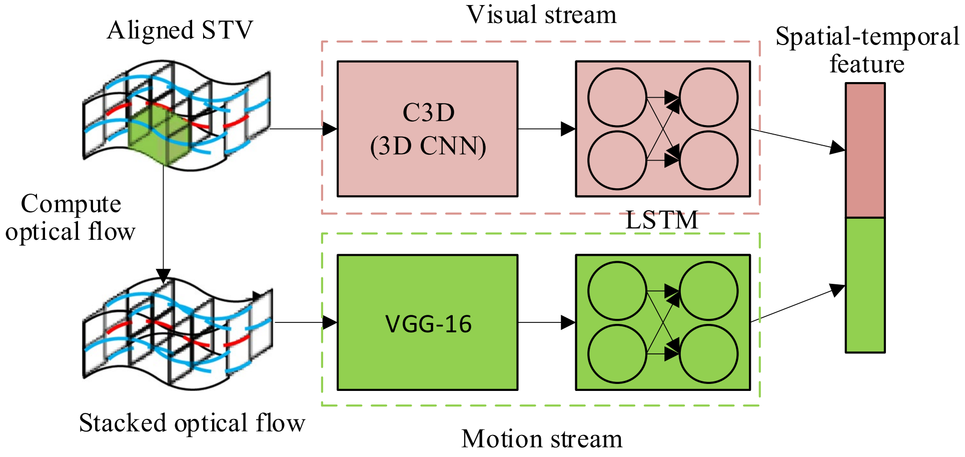


Fig. 7 The improved CNN-RNN based dual-stream network architecture. The visual stream (on top) is design by combing 3D CNN and LSTM to extract "appearance" context from raw pixels of STV data, while the motion (optical flow) stream (on bottom) learns motion information from optical flows by the CNN-LSTM scheme

of X can be modelled by a probability density function $p(u;\theta)$ with parameters θ, the X is described by the gradient vector:

$$G_\theta^X = \frac{1}{T}\nabla_\theta \log p(X;\theta) \tag{9}$$

The length of the gradient vector is fixed, which only depends on the number of parameters (i.e., the dimensionality of θ), but not the actual number of low-level features. The probability density function is widely used by models such as GMM: $p(u;\theta) = \sum w_i u_i(x)$, and $\theta = \{\pi_i, u_i, \sigma_i, i \in [1, K]\}$, where π_i, u_i and σ_i are the mixture weights, mean vector and diagonal GMM respectively, K denotes the mixture number of GMM. Then the fisher vector is formulated as follows:

$$\begin{aligned} g_{u,k}^X &= \frac{1}{T\sqrt{\pi_k}} \sum_{t=1}^{T} \gamma_t(k)\left(\frac{X_t-\mu_k}{\sigma_k}\right) \\ g_{\sigma,k}^X &= \frac{1}{T\sqrt{2\pi_k}} \sum_{t=1}^{T} \gamma_t(k)\left[\frac{(X_t-\mu_k)^2}{\sigma_k^2} - 1\right] \end{aligned} \tag{10}$$

where $\gamma_t(k)$ indicates the weight of low-level feature x_i for the j-th Gaussian function, as shown in the following:

$$\gamma_t(k) = \frac{\pi_k \mu_k(x_t)}{\sum_{j=1}^{K} \pi_j \mu_j(x_j)} \tag{11}$$

where $\mu_j(x_j)$ is D-dimensional Gaussian distribution, then the fisher vector of the set of features is given by the concatenation of $g_{u,k}^x$ and $g_{\sigma,k}^x$, shown as the follows:

$$f_{Fisher} = [(g_{u,k}^x)^{\mathrm{T}}, (g_{\sigma,k}^x)^{\mathrm{T}}]^{\mathrm{T}}, k \in [1, K] \tag{12}$$

Fisher vector encodes the average first and second order differences between the features and the centres of a GMM, which can be considered as a soft visual vocabulary demonstrating better performance than the bag of feature method for classification. To optimize the runtime performance of the design, the Principal Component Analysis (PCA) technique was first applied to reduce the low-level feature dimensionality. The number of Gaussians was set at $K = 512$ to train and estimate the GMM. Therefore a single video event can be represented by a $2DK$ dimensional fisher vector (see (12)) before a L2-normalization.

4.2 BOF-based representation

Bag of Feature (BOF) was inspired by the Bag of Words (BOW) model, and it is often referred as bag-of-visual-words (BOVW) in computer vision studies. In this case, a feature of an image or a video frame is considered as a "visual word".

The first stage of BOF implementation is to train the codebook. All low-level features extracted from training videos are clustered into N categories by the K-means clustering scheme. Such that each centre of a quantized area of a category becomes a visual word, and all visual words (cluster centres) construct the corresponding codebook. Thus, the length of a codebook is equal to the number of visual words in this codebook.

In the histogram calculating stage, the low-level features extracted from a video are represented as the histograms of visual words, denoted as:

$$C = (c_1, c_2, ..., c_n) \tag{13}$$

where c_i indicates the value of i-th visual word in the codebook. The value of c_i is normalized by the maximum-minimum functions:

$$n_i = \frac{c_i - \min(C)}{\max(C) - \min(C)} \tag{14}$$

Hence, a video event can be represented as the following histogram of visual words:

$$V = (n_1, n_2, ..., n_N) \tag{15}$$

BOF directly assigns a feature to one of the nearest visual words. This "hard" assignment is rigid and inaccurate. It is more flexible to assign a feature to different visual word bins when the distances between the feature and these visual words can be "weighted". Moreover, a feature may be assigned into different visual words when the scale of codebooks can be varied.

This study has applied a "soft-assignment" approach to rectify the aforementioned disadvantages based on the multi-assignment (MA) technique that can "split" a feature into multiple visual words [1]. In this case, a top-*N* nearest visual words method is devised for computing the weights for each visual word, and then the weights for a complete video sequence can be calculated as:

$$u_k = \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{M_i} \frac{1}{2^{i-1}} sim(j,k), k \in [1,K] \tag{16}$$

where u_k indicates the weight of *k*-th visual word, M_i describes the number of features whose *i*-th nearest neighbour is the visual word *k*, and function *sim*(*j*, *k*) calculates the similarity between the feature *j* and visual word *k*. Generally speaking, $N = 4$ achieves a good result based on the previous work [1]. Finally, a video event can be represented by the vector $TV = [u_1, u_2, ..., u_K]$.

4.3 BoTF formulation

As stated earlier, a BOF encodes a video event as a set of unordered local features. As a result, it struggles to deal with the temporal sequences of features, which could lead to problems in distinguishing "longer" or varied actions that constituting similar atomic components (but in different orders such as the motions of stand up and sit down). To address this issue, this study has devised a new feature representation method: Bag-of-Temporal-Features (BoTF) that embeds temporal information into BoF representation by employing the visual word correlograms and a co-occurrence transaction (CoTrans) scheme [10]. A correlogram not only contains the global spatial feature distribution of a video frame, but also has the corresponding spatial and temporal information encapsulated together [1]. Moreover, in this design, the CoTrans template has been applied to form feature patterns and to calculate the BoTF instances.

As a live implementation strategy, DT produces a set of low-level feature vectors $V = v_i$, where *v* represents a low-level feature of a video event (see Section 3). To explore the temporal information, this study introduced the time information into *v* so the feature is extended as [*t*, *v*] , where *t* indicates the time coordinate. In particularly, *t* is the time centre belonging to its trajectory. All features of a video event are ordered by temporal sequences (frame indexes), see Fig. 8 . Under the proposed system, the sequence for an event in a given time range *l* is denoted as: $PT = [tc, l, ori, v]$, where *tc* is the time centre, $l = 30$ denotes the number of frames in time-axis for the corresponding patch, $ori = \pm 1$ represents the orientation of polar axis, so a patch is defined as:

$$PT(t_c, l, ori, v) = \{[t_c, l, ori, v_i]\}, v_i \in V \tag{17}$$

And then, the CoTrans template is applied to calculate the BoTF instances based on all defined feature patches. The histogram *h*(*tc*, *l*, *ori*) encodes features in a feature patch *PT* through calculating the number of every visual word in *PT*. It is defined as the following:

$$h(t_c, l, ori) = (c_1, c_2, ..., c_k) \tag{18}$$

where *k* is the length of the codebook and c_i is the number of features in patch $PT(t_c, l, d)$ belonging to the *i*-th visual word.

In this step, similar to BoF, all low-level features extracted from a training dataset are clustered by using *K*-means for generating the codebook (a visual word set) of BoTF, and the vector length of $h(t_c, l)$ is equal to the length of the codebook generated by BoF. Moreover, the radial axis (*R*) is divided into $N^r = 4$ bins (N^r is equal to the number of feature patches on a quadrant of radial axis), the length of *R* is 60 frames and the polar axis ($\pm th$) is divided into $N^{\pm th} = 2$ bins ($N^{\pm th}$ is equal to the number of orientations of the polar axis), see Fig. 8 . Finally, the BoTF descriptor in accordance with CoTrans template can be formulated as the following:

$$BoTF_{tc} = [h(t_c, l, 1)_1, h(t_c, l, -1)_1, ..., h(t_c, l, 1)_4, h(t_c, l, -1)_4] \tag{19}$$

The set of CoTrans reference time centre is denoted as $C = \{t_1, t_2, ..., t_n\}$ that are sampled from time-axis by the successive 30 frames. With the BoTF descriptor, input live video streams can be represented as a set of $BoTF_{tc}$ descriptor instances. In conclusion, a video event is first described as a histogram of BoTF based visual words, and then (16) will be applied to assign a $BoTF_{tc}$ into multiple visual words.

4.4 Human action classification

The feature fusion strategy developed in this work enabled robust human action classification through a SVM based classifier. Three feature vectors (f_{Fisher}, BOF and $BoTF$) derived from the aforementioned models are fused into a final holistic video representation:

$$fv = [\lambda_1 f_{Fisher,} \lambda_2 BOF, \lambda_3 BoTF] \tag{20}$$

where λ_i $(i = 1, 2, 3)$ indicates the weight of each feature vector. This study considers all feature representations of equally weighting with a normalized $\lambda_i = 1$, so a video event can be represented as the holistic feature vector in real-time: $[f_{Fisher}, BoF, BoTF]$.

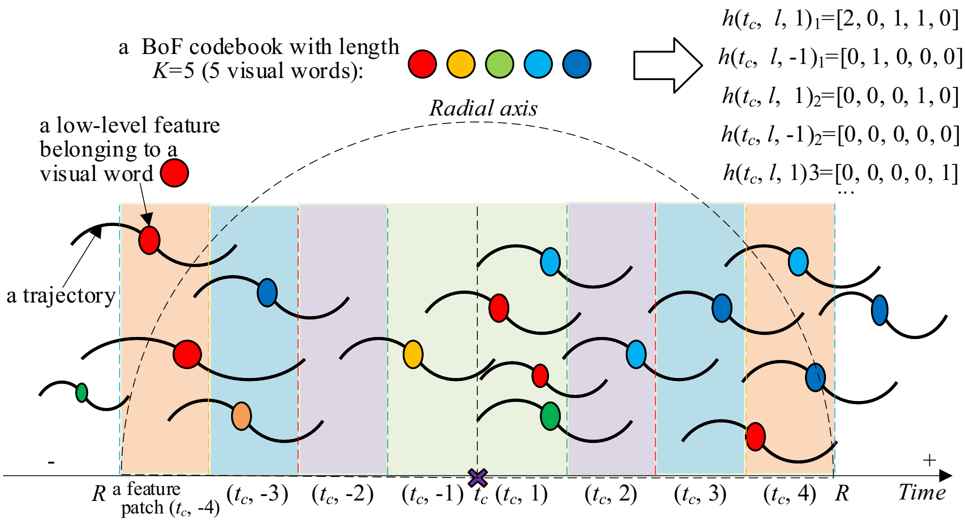

a BoF codebook with length
K=5 (5 visual words):
$h(t_c, l, 1)_1$=[2, 0, 1, 1, 0]
$h(t_c, l, -1)_1$=[0, 1, 0, 0, 0]
$h(t_c, l, 1)_2$=[0, 0, 0, 1, 0]
$h(t_c, l, -1)_2$=[0, 0, 0, 0, 0]
$h(t_c, l, 1)3$=[0, 0, 0, 0, 1]
...
Radial axis
a low-level feature belonging to a visual word
a trajectory
-
+
R
a feature patch (t_c, -4)
(t_c, -3)
(t_c, -2)
(t_c, -1)
t_c
(t_c, 1)
(t_c, 2)
(t_c, 3)
(t_c, 4)
R
Time

Fig. 8 Producing $BoTF_tc$ instances based on the BoF and the CoTrans templates

SVM is the optimal choice for dealing with relatively small sizes of handcrafted feature datasets. Thus, in order to test and evaluate the validity and efficiency of the devised framework, this study investigated SVM based classifier by comparing its performance when handling handcrafted, learned, and the combined features respectively. To classify multiple categories of actions, multi-SVM units have been generated, and each of them performs the "one-versus-the-rest" multi-class evaluation.

5 Experimental setup and evaluation

5.1 Dataset

The proposed human action analysis framework and its corresponding components and processes have been tested on 6 public benchmarking datasets, i.e., UCF 11, UCF 50, UCF 101, HMDB51, JHMDB51 and UT-Interaction datasets. Samples from those datasets are shown in Fig. 9.

The **UCF 11** dataset is an annotated version of YouTube clip collections [17]. It includes 11 individual actions namely, basketball shooting, cycling, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. UCF 50 is an extension of UCF 11 that contains 50 action categories collected from YouTube, while UCF 101 is also an extension of UCF 11 that has 13,320 videos from 101 action categories.

The **HMDB51** dataset [12] has been collected from movies and YouTube videos, and there are up to 51 action classes, while JHMDB51 is a sub-dataset based on 928 video clips from HMDB51 comprising 21 action categories.

The **UT-Interaction** [27] contains six types of human-human interactions: shake-hands, point, hug, push, kick and punch. The videos are divided into two sets of different environment settings.

5.2 Training strategy

Deep learning models require huge amount of labelled data for moulding the neural networks. However, many popular online action datasets are not adequate for this task. It is especially a challenge when handling real applications where datasets are often referring to noisy and untrimmed surveillance videos. As a result, many deep learning methods only achieved a low performance that even worse than the shallow handcrafted representations. Transfer learning suggests a significant advancement to utilise and be benefitted from small datasets [49], i.e., through training an initial network from scratch on a very large dataset (e.g., an ImageNet-like dataset) and then fine-turning the model on a task-specific dataset. However, the datasets used in this study are different from ImageNet. Directly applying transfer learning in this case will cause the underfitting problem. Motivated by this analysis, this study developed a multi-stage training strategy based on transfer learning. A public CNN model (e.g., VGG-16) pre-trained on the ImageNet ILSVRC-2014 dataset was adopted as the initial network. This pre-trained VGG models can be derived from online model repositories such as PyTorch Hub [32]. Then the model is fine-tuned on a relatively large dataset (UCF 101 action dataset) to ensure the robustness of the trained network. UCF 101 dataset supplies sufficient videos to fine-tune the entire network from image classification to motion analysis.

For training 3D CNN model, this study applied the same parameter settings in accordance with [35], and the C3D network was trained directly by using UCF 101 video clips.

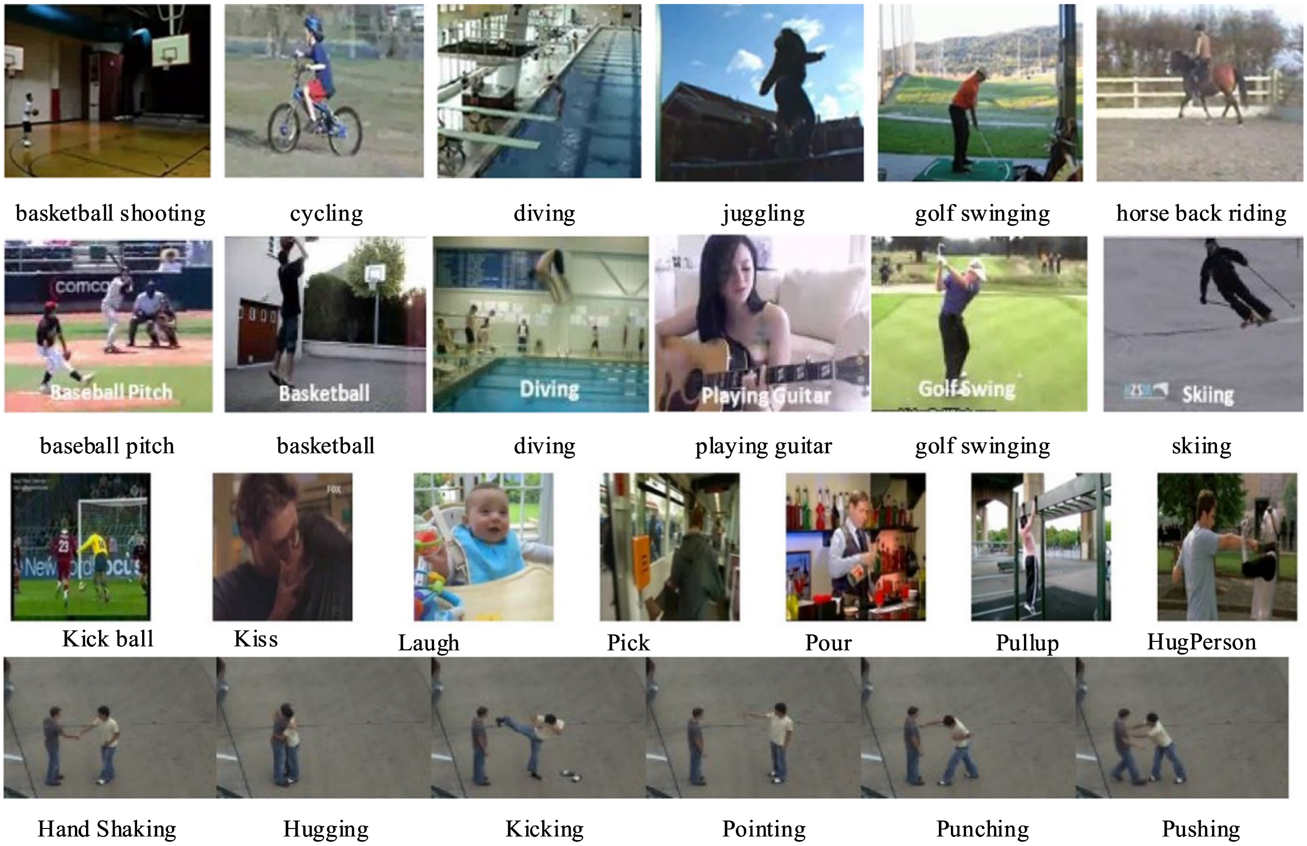


Fig. 9 Sample video clips from the four action datasets used in this study. From top to bottom: UCF11, UCF50, HMDB51 and UT-Interaction

The developed CNN models (i.e., VGG and C3D) are used as general feature extractors, whilst temporal features are identified through trained LSTM-based sequence models. In practice, the features from the last fully-connected layer are fed into LSTM units with M inputs $< x_1, x_2, ..., x_M >$ and M outputs $< y_1, y2, ..., y_M >$, where x_i presents a feature vector and y_i is the corresponding action label. The learnable weights (WR) of the LSTM-based sequence components can then be optimized by maximizing the likelihood of the ground truth outputs y_t calculated on the input data and the action labels. For a given training sequence $(x_m, y_m)_{m=1}^{M}$, this study minimizes the negative log likelihood: $L(WR) = -\sum_{m=1}^{M} \log P_{WR}(y_m|x_{1:m}, y_{1:m-1})$ by using SGD (stochastic gradient descent) with backpropagation algorithm to compute the gradient of the objective L with respect to the weights (WR).

Handcrafted feature extractors are not trainable methods, so the approaches discussed in Sections 3.1 - 3.3 to extract low-level handcrafted features were implemented. In this study, a dataset splits into three parts, i.e., the training subset (70%), the validation subset (10%) and the test subset (20%). Cross-validation strategy has been applied to train the SVM-based classifier to ensure the experiment accuracy and repeatability.

5.3 Hybrid feature descriptor efficiency

To evaluate the effectiveness of different feature strategies, this study compared various combinatory feature descriptors involving TS, HOG, HOF, MBH, and the learned features. Low-level features were extracted using techniques discussed in Section 3.3. The improved dual-stream CNN-RNN model that assembles C3D, VGG-16 and LSTM networks were adopted to extract learned features as highlighted in Section 3.4. The combined video representation based on the FV and BoTF methods was applied to encode a video event as a holistic feature vector that was then gone through a SVM classifier for action classification.

In this experiment, the UCF 50 dataset was used, and the result is shown in Table 1. It can be seen that a single TS feature generally shows weaker performance, while the MBH achieves the best accuracy rate among those individual features due to the MBH descriptor focuses on tracking human foreground motions, whilst the camera motions and background change are removed. Learned features, on the other hand, gains better performance than TS and HOG but not levelling with HOF and MBH, which is explainable due to the nature of a pre-trained generic deep feature model

Table 1 The recognition accuracy rate of different feature sets on UCF 50

Features	Accuracy (%)
TS	76.5
HOG	83.6
HOF	87.5
MBH	90.3
Handcrafted (TS+HOG+HOF+MBH)	92.5
Dual-stream (VGG-16 & C3D)	89.6
Hybrid Model	**95.68**

- not a specifically trained one - was adopted in this work. Unsurprisingly, the hybrid feature descriptor demonstrates the best performance by harnessing the advantages from underlying feature types with the recognition rate reaching 95.68%. It is observed that the overall performance of the hybrid model increased by over 5% when compared with MBH and the original dual-stream framework, respectively. One possible reason is that certain actions in UCF 50 videos have more salient motion information (e.g., TaiChi and High Jump actions) while other actions possess less distinctive motions within different scenes, e.g., biking and horse-riding. An individual feature descriptor (e.g., MBH) can extract motion information on the former type of action, it often falls short in handling actions of the latter type and resulting the loss of the overall accuracy. In contrast, when applying the hybrid model, both motion and scene information can be extracted more thoroughly, so as to improve recognition rates for wider spectrum of action types.

As shown in Table 1, the performance of the combined handcrafted features (i.e., the combinations of TS, HOG, HOF and MBH) is better than any individual one, which indicates the relevance of all aspects of handcrafted features towards the final prediction results. Based on this observation, this study integrated the combined handcrafted features for the rest of the work. Moreover, this study also tested the performance of handcrafted and learned features separately to gain insight of their respective impacts to the outcome. The hybrid model has further demonstrated its superiority over the single-stream-based approach on all tested benchmarks drawn from ablation studies.

5.4 Dual-stream architecture comparison

This study also examined different CNN architectures in dual-stream deep learning models (see the CNN components in Fig. 7, which can be implemented by different CNN architectures) to identify a suitable one for the devised framework. Four popular CNN models for image classification were implemented to extract video features, namely, AlexNet [11], VGG-16, VGG-19 and the C3D network. The former three CNN models are pre-trained by ImageNet image classification dataset [11] and the C3D was trained by UCF 101 dataset. Then, the optimized FV and BoTF video representation scheme were applied. In this experiment, the UCF 50 dataset was used to test these implementations. It is clearly shown in Table 2 that the accuracies of AlexNet for both streams are lower than the VGG model, while the performance of VGG-16 is identical to VGG-19. However, VGG-19 requires more computational resources than VGG-16 due to its the extra network depth. Hence, parameters from the pre-trained VGG-16 are inherited as the generic learned feature extractor to archives the best accuracy-cost trade-offs. It is worth noting that the performance improvement is significant when adapting C3D network in the visual stream, and the main reason is that the 3D CNN used in C3D is more effective when extracting spatial-temporal features from STV data. However, the accuracy does not improve significantly when adopting the C3D network in the motion stream, one of the main factors is that C3D mainly focuses on capturing high level abstract and semantic information from RGB video clips, while optical flow only abstracts motion information. According to the performance comparison including processing time, this study has adopted the C3D and VGG-16 networks to implement the transferred feature extractors.

This experiment compared the dual-stream model with the individual stream settings (i.e., using either visual or motion stream), and the result is shown in Table 2. Unsurprisingly, the dual-stream model performed consistently better than the singe-stream settings. According to the experimental result, this study adopted the C3D and VGG-16 configurations in the dual-stream network for the further experiments.

5.5 Event representation validation

The performance of iDT (based on fisher vector) [41], stacked fisher vector (SFV) [24], BoTF, and the proposed approach (a combination of FV and BoTF) had been applied on the UCF11 and UCF50 datasets for evaluation. Table 3

Table 2 The recognition accuracy rates of different network architectures for the CNN components of the dual-stream deep learning models on UCF 50 dataset

Deep feature model		
Spatial stream	Temporal stream	Accuracy (%)
AlexNet	AlexNet	76.4
VGG-16	VGG-16	85.6
VGG-19	VGG-19	85.8
C3D	**VGG-16**	**89.6**
C3D	VGG-19	89.8
C3D	C3D	86.4
C3D	-	85.2
-	VGG-16	79.5

Table 3 Accuracy rates of individual actions using different trajectory models

Method	UCF11/YouTube (%)	UCF50 (%)
iDT (FV)	90.69	90.5
SFV	93.38	NA
BoTF	92.5	91.3
FV+BoTF (Hybrid Model)	**95.40**	**95.68**

illustrates the recognition accuracy variation of these four models. It is shown that the performance of SFV is superior to the traditional iDT model since the SFV encodes the sematic information in a hierarchical structure. It is worth mentioning that pure BoTF implementation is also slightly better than iDT but does not gain extra over SFV. One possible reason is that the BoTF representation encodes the features in different time patches, and it can describe the temporal sequences of features, so as to handle the "longer" and varied actions. However, the BoTF mainly focuses on temporal information and containing less spatial information than FV and SFV. Unsurprisingly, when combined with FV, it achieves the best performance on tested UCF11 and UCF50 instances. The superiority is rooted in the presence of both local and global features over the spatial and temporal domains.

5.6 Comparison with the state-of-the-art in action recognition

This study has compared the proposed hybrid model with other state-of-the-art methods on UCF 101, HMDB51 and JHMDB51 datasets, including iDT [41], SFV [24], the dual-stream model proposed by Simonyan et al. [29], and its improved versions, namely a few, the hidden two-stream [56], two-stream with LSTM models [4], the two-in-one stream model [53], the long-term convolution networks (LTC) [37], the C3D model [35], the C^{2}LSTM model [21], and the most recent LSF CNN model [39]. Evaluations have also been carried out in comparison with the hybrid handcrafted and learning-based methods, such as the trajectory-pooled deep convolutional descriptors (TDD) [43], and the MTC3D model released in 2019 [20]. HMDB51 and JHMDB51 datasets are more complex than UCF datasets in terms of action types, video quality, and background. Experiments show a superior output from the devised hybrid model in this research as highlighted in Table 4. The configurations of the testbed are detailed in Section 5.3. The performance of the devised hybrid model is consistently level up or surpass current benchmark approaches. The superior performance stems into the trajectory-based (handcrafted) features among multiple scales, separable frequency bands and directions, and the fine-grained deep learned spatial-temporal features.

5.7 Applicability and extensibility

To investigate and evaluate the generalization of the proposed hybrid model, this study also tested extended human action categories such as those depicted in UT-Interaction dataset that mainly focuses on human-human interactions. The same configuration settings as described in Section 5.3 has been adopted for the test. Since videos in this series (set1 and set2) contain combinatory actions, segmented datasets were deployed in this experiment. Comparing to the BoF [26] with the deep representation proposed by Lee et al. in 2019 [15], the devised framework demonstrated robustness and greatly extended applicability on complex human interactions evidenced by the state-of-the-art performances shown in Table 5. Overall, with the holistic features, the proposed model has gained significant performance advancements on both human actions and interactions with convincing promise on crowd action understanding.

Table 4 Performance comparison to the state-of-the-art approaches on UCF 101, HMDB51 and JHMDB51

Method	UCF101 (%)	HMDB51 (%)	JHMDB51 (%)
iDT (2013)	87.9	57.2	62.8
SFV (2014)	NA	66.79	69.03
C3D (2015)	85.2	NA	NA
TDD (2015)	90.3	63.2	NA
LTC (2018)	91.7	64.8	NA
Dual-stream CNN (2014)	88.0	59.4	NA
Two-stream + LSTM (2017)	88.6	NA	NA
Hidden Two-stream (2019)	90.3	60.5	NA
Two-in-one (2019)	92.0	NA	45.01
C^{2}LSTM (2020)	92.8	61.3	NA
MTC3D (2019)	90.1	65.0	NA
SLF CNN (2020)	94.8	70.2	NA
Hybrid Model	**95.1**	**70.8**	**72.5**

Table 5 Extensibility on UT-Interaction dataset

Method	Set #1 (%)	Set #2 (%)
BOF	81.67	80.00
deep representation	90.22	89.40
Hybrid Model	**91.35**	**91.50**

6 Conclusions and future work

To tackle the shortfalls of lacking orientations and separable frequencies in multiple scales of traditional DT-based action classification models, this study has developed an innovative DT model by integrating the DWT technique. 2D DWT method is employed to decompose the video frame into separable frequency and orientation components for abstracting motion information. Dense trajectories method is applied to extract feature points for tracking through consecutive frames. A hybrid framework integrates both handcrafted and learned features for harnessing their distinctive characteristics over the spatial and temporal spaces. Fisher vector and a novel handcrafted feature representation - BoTF, have been developed to encode video events. The holistic representation of video-based events over time, specifically human actions in this research, enables efficient and accurate analysis through classification. Experiments show that the proposed hybrid model has superior recall, robustness, and extensibility performance over benchmarked systems and approaches.

Both handcrafted and deep learning methods were thoroughly examined in this study, which open up a significant research direction through harnessing advantages from different feature engineering paradigms. This study also tested traditional feature fusion solutions such as the score-based model, which was proven inefficient. Currently, the multikernel learning and the metric-based learning approaches are the most popular ones to fuse features, which reveal the potential of internal network fusion through and at CNN feature maps. One limitation of this study is that the devised human action recognition system can only process trimmed video clips with clear action boundaries, such as those from the UCF and HMDB datasets. A number of aspects have been explored as a preparation for the follow ups including tailored deep learning networks and adaptive feature weighting to better handle varied lengths of ambiguous crowd behavioural events.

Acknowledgements This research is supported by the National Natural Science Foundation of China (NSFC) (61203172); the Sichuan Science and Technology Programs (2019YFH0187, 2020018); and the European Commission (598649-EPP-1-2018-1-FR-EPPKA2-CBHE-JP).

References

1. Bolovinou A, Pratikakis I, Perantonis S (2013) Bag of spatio-visual words for context inference in scene classification. Pattern Recognition 46(3):1039–1053. https://doi.org/10.1016/j.patcog.2012.07.024
2. Chandra MA, Bedi SS (2018) Survey on SVM and their application in image classification. International Journal of Information Technology pp 1–11. https://doi.org/10.1007/s41870-017-0080-1
3. Chang J, Wang L, Meng G, Xiang S, Pan C (2017) Deep Adaptive Image Clustering. In: International Conference on Computer Vision. IEEE, pp 5880–5888. https://doi.org/10.1109/ICCV.2017.626
4. Gammulle H, Denman S, Sridharan S, Fookes C (2017) Two Stream LSTM: A Deep Fusion Framework for Human Action Recognition. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp 177–186. IEEE. https://doi.org/10.1109/WACV.2017.27
5. Ji S, Xu W, Yang M, Yu K (2013) 3D Convolutional Neural Networks for Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1):221–231. https://doi.org/10.1109/TPAMI.2012.59
6. Jiang J, Deng C, Cheng X (2017) Action prediction based on dense trajectory and dynamic image. In: Chinese Automation Congress. IEEE, pp 1175–1180 https://doi.org/10.1109/CAC.2017.8242944
7. Jin S, Su H, Stauffer C, LearnedMiller E (2017) End-to-End Face Detection and Cast Grouping in Movies Using Erdös-Rényi Clustering. In: International Conference on Computer Vision. IEEE, pp 5286–5295 https://doi.org/10.1109/ICCV.2017.564
8. Ju S, Xiao W, Shuicheng Y, LoongFah C, Tat-Seng C, Jintao L (2009) Hierarchical spatio-temporal context modeling for action recognition. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp 2004–2011 https://doi.org/10.1109/CVPRW.2009.5206721
9. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L(2014) Large-Scale Video Classification with Convolutional Neural Networks. In: Computer Vision and Pattern Recognition. IEEE, pp 1725–1732 https://doi.org/10.1109/CVPR.2014.223
10. Kieu T, Vo B, Le T, Deng ZH, Le B (2017) B: Mining top-k co-occurrence items with sequential pattern. Expert Systems with Applications 85:123–133. https://doi.org/10.1016/j.eswa.2017.05.021
11. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Communications of the ACM 60(6):84–90. https://doi.org/10.1145/3065386
12. Kuehne H, Jhuang H (2011) HMDB: A large video database for human motion recognition. In: International Conference on Computer Vision. IEEE, pp 2556–2563 https://doi.org/10.1109/ICCV.2011.6126543
13. Laptev I (2005) On Space-Time Interest Points. International Journal of Computer Vision 64(2):107–123. https://doi.org/10.1007/s11263-005-1838-7
14. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 1–8 https://doi.org/10.1109/CVPR.2008.4587756
15. Lee DG, Lee SW (2019) Prediction of partially observed human activity based on pre-trained deep representation. Pattern Recognition 85:198–206. https://doi.org/10.1016/j.patcog.2018.08.006
16. Li W, Wen L, Chang M, Lim SN, Lyu S (2017) Adaptive RNN Tree for Large-Scale Human Action Recognition. In: International Conference on Computer Vision. IEEE, pp 1453–1461 https://doi.org/10.1109/ICCV.2017.161
17. Liu J, Jiebo Luo, Shah M (2009) Recognizing realistic actions from videos “in the wild”. In: Conference on Computer Vision

and Pattern Recognition. IEEE, pp 1996–2003 https://doi.org/10.1109/CVPR.2009.5206744

18. Liu P, Wang J, She M, Liu H (2011) Human action recognition based on 3D SIFT and LDA model. In: Workshop on Robotic Intelligence In Informationally Structured Space, pp 12–17 https://doi.org/10.1109/RIISS.2011.5945790
19. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg AC (2016) SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision, pp 21–37 https://doi.org/10.1007/978-3-319-46448-0_2
20. Lu X, Yao H, Zhao S, Sun X, Zhang S (2019) Action recognition with multi-scale trajectory-pooled 3D convolutional descriptors. Multimedia Tools and Applications 78(1):507–523. https://doi.org/10.1007/s11042-017-5251-3
21. Majd M, Safabakhsh R (2020) Correlational Convolutional LSTM for human action recognition. Neurocomputing 396:224–229. https://doi.org/10.1016/j.neucom.2018.10.095
22. Messing R, Pal C, Kautz H (2009) Activity recognition using the velocity histories of tracked keypoints. In: International Conference on Computer Vision. IEEE, pp 104–111 https://doi.org/10.1109/ICCV.2009.5459154
23. Murtaza F, HaroonYousaf M, Velastin SA (2018) DA-VLAD: Discriminative Action Vector of Locally Aggregated Descriptors for Action Recognition. In: IEEE International Conference on Image Processing (ICIP). IEEE, pp 3993–3997 https://doi.org/10.1109/ICIP.2018.8451255
24. Peng X, Zou C, Qiao Y, Peng Q (2014) Action Recognition with Stacked Fisher Vectors. In: European Conference on Computer Vision. Springer, pp 581–595 https://doi.org/10.1007/978-3-319-10602-1_38
25. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You Only Look Once: Unified, Real-Time Object Detection. In: Computer Vision and Pattern Recognition. IEEE, pp 779–788 https://doi.org/10.1109/CVPR.2016.91
26. Ryoo MS (2011) Human activity prediction: Early recognition of ongoing activities from streaming videos. In: International Conference on Computer Vision. IEEE, pp 1036–1043 https://doi.org/10.1109/ICCV.2011.6126349
27. Ryoo MS, Aggarwal JK (2010) UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). https://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html
28. Sargano A, Angelov P, Habib Z (2017) A Comprehensive Review on Handcrafted and Learning-Based Action Representation Approaches for Human Activity Recognition. Applied Sciences 7(1):110. https://doi.org/10.3390/app7010110
29. Simonyan K, Zisserman A (2014) Two-Stream Convolutional Networks for Action Recognition in Videos. In: Advances in neural information processing systems, pp 568–576
30. Simonyan K, Zisserman A (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations, pp 769–784
31. Sipiran I, Bustos B (2011) Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. In: Visual Computer, p. 963–976 https://doi.org/10.1007/s00371-011-0610-y
32. Steiner B, DeVito Z, Chintala S, Gross S, Paszke A, Massa F, Lerer A, Chanan G, Lin Z, Yang E, Desmaison A, Tejani A, Kopf A, Bradbury J, Antiga L, Raison M, Gimelshein N, Chilamkurthy S, Killeen T, Fang L, Bai J (2019) PyTorch: An Imperative Style. Advances in Neural Information Processing Systems (NIPS), High-Performance Deep Learning Library. In
33. Sun J, Mu Y, Yan S, Cheong LF (2010) Activity recognition using dense long-duration trajectories. In: International Conference on Multimedia and Expo. IEEE, pp 322–327 https://doi.org/10.1109/ICME.2010.5583046
34. Tao M, Bai J, Kohli P, Paris S (2012) SimpleFlow: A non-iterative, sublinear optical flow algorithm. Computer Graphics Forum 31(2):345–353. https://doi.org/10.1111/j.1467-8659.2012.03013.x
35. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning Spatiotemporal Features with 3D Convolutional Networks. In: International Conference on Computer Vision, 1. IEEE, pp 4489–4497 https://doi.org/10.1109/ICCV.2015.510
36. Van Droogenbroeck M, Barnich O (2014) ViBe: A Disruptive Method for Background Subtraction. In: Background Modeling and Foreground Detection for Video Surveillance. Chapman and Hall/CRC, pp 7.1–7.23 https://doi.org/10.1201/b17223-10
37. Varol G, Laptev I, Schmid C (2018) Long-Term Temporal Convolutions for Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(6):1510–1517. https://doi.org/10.1109/TPAMI.2017.2712608
38. Vishwakarma DK, Kapoor R (2015) Hybrid classifier based human activity recognition using the silhouette and cells. Expert Systems with Applications 42(20):6957–6965. https://doi.org/10.1016/j.eswa.2015.04.039
39. Wan Y, Yu Z, Wang Y, Li X (2020) Action Recognition Based on Two-Stream Convolutional Networks with Long-Short-Term Spatiotemporal Features. IEEE Access 8:85284–85293. https://doi.org/10.1109/ACCESS.2020.2993227
40. Wang H, Kläser A, Schmid C, Liu C (2013) Dense Trajectories and Motion Boundary Descriptors for Action Recognition. International Journal of Computer Vision 103(1):60–79. https://doi.org/10.1007/s11263-012-0594-8
41. Wang H, Schmid C (2013) Action Recognition with Improved Trajectories. In: International Conference on Computer Vision. IEEE, pp 3551–3558 https://doi.org/10.1109/ICCV.2013.441
42. Wang J, Xu Z (2013) STV-based video feature processing for action recognition. Signal Processing 93(8):2151–2168. https://doi.org/10.1016/j.sigpro.2012.06.009
43. Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep-convolutional descriptors. In: Computer Society Conference on Computer Vision and Pattern Recognition, pp 4305–4314 https://doi.org/10.1109/CVPR.2015.7299059
44. Wang Y, Long M, Wang J, Yu PS (2017) Spatiotemporal Pyramid Network for Video Action Recognition. In: Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2097–2106 https://doi.org/10.1109/CVPR.2017.226
45. Wu G, Mahoor MH, Althloothi S, Voyles RM (2010) SIFT-Motion Estimation (SIFT-ME): A New Feature for Human Activity Recognition. In: IPCV, pp 804–811
46. Wu W, Kan M, Liu X, Yang Y, Shan S, Chen X (2017) Recursive Spatial Transformer (ReST) for Alignment-Free Face Recognition. In: International Conference on Computer Vision. IEEE, pp 3792–3800 https://doi.org/10.1109/ICCV.2017.407
47. Xue F, Zhang W, Xue F, Li D, Xie S, Fleischer J (2021) A novel intelligent fault diagnosis method of rolling bearing based on two-stream feature fusion convolutional neural network. Measurement 176:109226. https://doi.org/10.1016/j.measurement.2021.109226
48. Yao G, Lei T, Zhong J, Jiang P (2019) Learning multi-temporal-scale deep information for action recognition. Applied Intelligence 49(6):2017–2029. https://doi.org/10.1007/s10489-018-1347-3
49. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14. MIT Press, Cambridge, MA, USA, p 3320–3328
50. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Springer, pp 818–833 https://doi.org/10.1007/978-3-319-10590-1_53

51. Zhang HB, Zhang YX, Zhong B, Lei Q, Yang L, Du JX, Chen DS (2019) A Comprehensive Survey of Vision-Based Human Action Recognition Methods. Sensors 19(5):1005. https://doi.org/10.3390/s19051005
52. Zhang J, Hu H (2019) Domain learning joint with semantic adaptation for human action recognition. Pattern Recognition 90:196–209. https://doi.org/10.1016/j.patcog.2019.01.027
53. Zhao J, Snoek CGM (2019) Dance With Flow: Two-In-One Stream Action Detection. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp 9927–9936 https://doi.org/10.1109/CVPR.2019.01017
54. Zhao L, Tang P, Huo L (2014) A 2-D wavelet decomposition-based bag-of-visual-words model for land-use scene classification. International Journal of Remote Sensing 35(6):2296–2310. https://doi.org/10.1080/01431161.2014.890762
55. Zhao LJ, Tang P, Huo LZ (2014) Land-use scene classification using a concentric circle-structured multiscale bag-of-visual-words model. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7(12):4620–4631. https://doi.org/10.1109/JSTARS.2014.2339842
56. Zhu Y, Lan Z, Newsam S, Hauptmann A (2019) Hidden Two-Stream Convolutional Networks for Action Recognition. In: Asian Conference on Computer Vision, pp 363–378 https://doi.org/10.1007/978-3-030-20893-6_23