



An attention mechanism based convolutional LSTM network for video action recognition

Hongwei Ge¹  · Zehang Yan¹ · Wenhao Yu¹ · Liang Sun¹

Received: 2 June 2018 / Revised: 2 February 2019 / Accepted: 22 February 2019 /
Published online: 2 March 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

As an important issue in video classification, human action recognition is becoming a hot topic in computer vision. The ways of effectively representing the spatial static and temporal dynamic information of videos are important problems in video action recognition. This paper proposes an attention mechanism based convolutional LSTM action recognition algorithm to improve the accuracy of recognition by extracting the salient regions of actions in videos effectively. First, GoogleNet is used to extract the features of video frames. Then, those feature maps are processed by the spatial transformer network for the attention. Finally the sequential information of the features is modeled via the convolutional LSTM to classify the action in the original video. To accelerate the training speed, we adopt the analysis of temporal coherence to reduce the redundant features extracted by GoogleNet with trivial accuracy loss. In comparison with the state-of-the-art algorithms for video action recognition, competitive results are achieved on three widely-used datasets, UCF-11, HMDB-51 and UCF-101. Moreover, by using the analysis of temporal coherence, desirable results are obtained while the training time is reduced.

Keywords Attention mechanism · Convolutional LSTM · Spatial transformer · Video action recognition

1 Introduction

Nowadays, much information on the Internet is communicated via multimedia data including texts, audio, images and videos. Among them, videos contain massive information. The increasing cameras and high-speed broadband networks keep videos growing more, resulting in the information explosion [15]. It is important to understand the content of a video for various applications in retrieval [55], recommendation [41], surveillance [46], human interaction prediction [53], virtual reality [21], etc. Considering that most videos focus on

✉ Hongwei Ge
hwge@dlut.edu.cn

¹ College of Computer Science and Technology, Dalian University of Technology,
Dalian 116023, China

human activities, it is desirable to recognize human actions accurately and robustly, which requires algorithms to extract discriminative features effectively.

A variety of methods based on traditional machine learning have been developed for video classification. Peng et al. proposed a model based on bag of visual words (BoVW) with local features for action recognition [29]. Bhattacharya et al. proposed a probabilistic representation for visual classification tasks based on maximizing the likelihood of generating the observed visual words [2], which is an efficient alternative to the traditional vocabulary based on bag-of-visual words. To employ more comprehensive information, Ikizler-Cinbis et al. consider the features of objects, scenes and people simultaneously in a multiple instance learning (MIL) framework [12]. Wang et al. extract video features based on dense trajectories and motion boundary descriptors [45]. Most of the improvements to visual analysis can be attributed to the introduction of improved feature extractors and feature encoding methods [7, 19, 28, 40].

Obtaining the remarkable success on image classification, 2D convolutional neural networks (CNN) are recently employed by [14, 25] to extract the features of each frame in a video. The frame features are then directly assembled to get the video feature. However, these methods are incapable of capturing the temporal dynamics. In order to solve this problem, in [6, 33, 37, 47, 52], some 2D CNN extensions are improved by either being extended to 3D CNN or manually designed to receive explicit temporal inputs (e.g. optical flow). As a natural way to capture temporal dynamics, recurrent neural network (RNN) is combined with CNN to extract video features, but the features in each spatial locality are treated fairly [5, 22, 43, 48, 49]. In [17], Krizhevsky et al. reported that when asked to classify an image, humans focus on the salient discriminative parts rather than the whole picture. After that, different attention-based methods were proposed and achieved promising results on several challenging tasks, including image caption generation [51], machine translation [1], game-playing and tracking [26], and image recognition [56]. Recently, Sharma et al. [32] and Li et al. [23] introduced visual attention mechanism in video recognition. The attention mechanisms can be grouped into two categories, soft deterministic and hard stochastic attention mechanisms for pooling convolutional feature maps. Since it requires sampling, training a hard attention based model incurs heavy computational burden. Requiring no sampling and less computation, the soft attention based model is adopted in [23, 32].

Simply weighing and summing all the spatial localities, soft attention might bring in noises. While hard attention focuses only one spatial locality, which might be too local to cover other possible discriminative features. Videos consisting of varying frames are much more complicated than images. Devised for static images and not considering the relevance between consecutive frames, the attention method (the spatial transformer) cannot be used for videos directly.

Inspired by the spatial transformer proposed in [13], we propose a novel attention based deep neural network which can dynamically sample multiple salient spatial localities in convolutional feature maps by affine transformation. For the first time, Long-Short Term Memory Model (LSTM) [11] is combined with the spatial transformer for video action recognition. Capable of selecting the discriminative localities by sampling different feature maps, the proposed LSTM spatial transformer leverages the relevance between consecutive frames, while consuming less computation than hard attention based methods and obtaining higher classification accuracy than soft attention. The analysis of temporal coherence in videos [31] is also incorporated to save the training time, making the runtime acceptable for training on large-scale datasets.

2 Related works

Recently, many end-to-end methods for video recognition based on deep learning have emerged. In this paper, our work is based on neural networks and attention mechanism. Thus in this section, we review some of the researches on convolutional neural networks, recurrent neural networks, and visual attention mechanisms involved in video recognition.

2.1 Convolutional neural networks for action recognition

The success of CNN extracting the deep features on image-analyzing tasks has inspired researchers for video classification. A practicable choice is to pre-train the CNN, such like Alexnet [17], Vggnet [34], GoogleNet [38] and ResNet [10] on ImageNet dataset [4] and use it as a feature extractor. Moreover, CNN combined with the independent subspace analysis can learn invariant spatio-temporal features from unlabeled video data [20]. Intuitively, a video can be taken as a set of frames, and the final representation of a video can be computed by averaging the feature vectors of all the frames, which are extracted by feeding a CNN one frame at a time. Obviously, taking a video as a set of pictures and directly feeding them to the CNN might lose sight of the important temporal dynamics in a video. Ji et al. [52] generalize 2D CNN to the 3D case by performing 3D convolution in both spatial and temporal dimensions. One convolution layer can capture the motion information from a few consecutive frames, and so forth, the motion information spanning across all the frames will be extracted ultimately. Karpathy et al. [15] evaluate different kinds of connectivity of the CNN in the temporal dimension and proposed a multi-resolution method to speed up the training. Tran et al. [41] train a 3D CNN on a large video dataset to obtain generic spatiotemporal features. For handling 3D signals more efficiently, Sun et al. factorize 3D convolution as a combination of 2D spatial convolution followed by 1D temporal convolution [37]. Similarly, Qiu et al proposed a new architecture named Pseudo-3D Residual Net (P3D ResNet) based on the factorization of 3D convolution [30]. However, 3D CNNs still face difficulties that their performance is currently not good enough [44]. The reason may be that too much noise in the temporal domain makes it hard to extract accurate motion information. To avoid direct convolution over the noisy temporal domain, Simonyan and Zisserman [33] proposed a two-stream architecture, which incorporates spatial and temporal networks. The spatial network is trained on static RGB frames to extract spatial information, and the temporal network is trained on dense optical flow to extract motion information. The dense optical flow is used to indicate the motion information explicitly. The algorithm makes the training easier and achieves competitive performance. Feichtenhofer et al. provide a number of ways of fusing two-stream networks to best take advantage of spatio-temporal information [6]. The important factors related to the performance of the two-stream CNN are investigated by Ye et al. [55], including network architectures, model fusion, learning parameters and the final prediction methods. Recently, Carreira et al. [3] proposed a two-stream inflated 3D-CNN (I3D) model that is based on 2D-CNN inflation. Further, the model is pre-trained on a new large-scale video dataset Kinetics and achieves extraordinary performance. Tran et al. [42] design the R(2+1)D model by using 3D-CNN factorization and integrate the model within the framework of residual learning. After pre-training on Kinetics, their model also obtains comparable results with [3]. Most methods use 2D or 3D CNNs to process videos as RGB image sequences, or employ optical flow for further boosting performance. Recently, Wu et al. [50] proposed an action recognition algorithm based on compressed video (CoViAR).

They design a multi-CNN structure trained directly on compressed videos to excavate the spatiotemporal features. CoViAR is the first to train deep neural networks on compressed videos and makes full use of the compressed information with higher density. CoViAR yields a new state-of-the-art, which indicates that compressed video coding would provide an effective way for video analysis.

2.2 Recurrent neural networks for action recognition

The temporal CNN in the two-stream method [33] can only explicitly capture the motion between consecutive frames and depict short-period actions. During the training phase, either a single frame or a multi-frame optical flow is fed into the CNN. Thus it has no consideration for the frame order. With the two characteristics, some longer-period actions consisting of several sub-actions cannot be processed effectively.

Different from CNN taking images or static frames of fixed length as inputs, RNN is a natural way designed for modeling temporal dynamics. Thus, researchers employed RNN to model the intricate temporal dynamics in videos. Exempted from the vanishing gradient problem, LSTM, among many RNN variants, has been shown to be effective in many tasks, such as image/video title generation [5, 54] and voice analysis [8]. Srivastava et al. obtain the representations of videos by using an encoder LSTM to map an input sequence to a fixed length representation [36]. Donahue et al. [5] proposed an end-to-end recurrent convolution architecture, which assembles the recurrent sequential model and the convolutional visual model directly. The convolutional part can be extended further to a two-stream architecture as in [33]. The sequential and visual models are trained simultaneously, and the spatial and temporal dynamics are learned respectively. Wu et al. [48] proposed a hybrid deep neural framework to model static spatial information, short-term motion and long-term dynamics. In this framework, the spatial and short-term features are extracted by two CNNs and are combined by using a feature-fusion network. Then, LSTM is employed to model the longer-term dynamics based on the fused features. Considering that actions can span across varying-length frames (individual frame, continuous frames, short segments or the entire video), Li et al. [22] proposed a framework to learn the deep spatial-temporal video representation in a hierarchical and multi-granular fashion. Ng et al. proposed several deep neural network architectures to combine frame information across a video over longer time periods [27]. Some works have contributed in mobile devices-based applications where human activity recognition is an important problem. Tao et al. present a new two-directional feature derived from horizontal and vertical acceleration components. Moreover, a multicolumn bidirectional LSTM (MBLSTM) ensemble classifier is proposed to combine different features for improving recognition accuracy [39]. Besides accelerometers, human actions can also be represented by the multiview features from depth and inertial sensors. Guo et al. encode the multiview features into a unified space that is optimal for activity recognition [9].

2.3 Visual attention mechanisms for video recognition

One frame contains the content beneficial for recognition but it also holds irrelevant noise. The useful content should be recognized to improve the accuracy of recognition and the useless content should be omitted to reduce extra computation. Sharma et al. [32] proposed a soft attention based LSTM model for action recognition, which learns to focus more on the relevant parts of a frame. However, similar to soft attention mechanism for image caption generation in [51], they did not consider the motion information during the attention procedure. Li et al. [23] introduced the VideoLSTM applying attention in LSTM models

by hardwiring convolution to the LSTM and adding motion for better attention, which are not only important for the action classification, but also result in better attention for action localization. More recently, Jaderberg et al. [13] introduced the spatial transformer, a new way to learn the invariance for generic warping by affine transformation. The spatial transformer can extract salient local information and reduce the side effects brought by noisy information. Yan et al. [53] proposed a tri-coupled LSTM structure embedded with a relative attention model for human interaction prediction. Different from the traditional attention selecting discriminative regions based on only one LSTM, the relative attention predicts discriminative regions by using the hidden states of three coupled LSTMs.

Motivated by the above reviews, this paper proposes an attention mechanism based convolutional LSTM network for video action recognition. In the proposed method, GoogleNet is employed to extract deep feature maps for video frames. The LSTM spatial transformer is proposed to overcome the disadvantages of soft and hard attention mechanisms by intrinsically considering both motion and space information. The convolutional LSTM module is then used to classify by integrating the spatial and temporal information of the feature maps of each video frame.

3 An attention mechanism based convolutional LSTM network

The proposed framework is an end-to-end model receiving video frames as input and outputting the final classification result for videos. The stacked network architecture for video classification can be divided into four stages: video feature extraction, salient feature selection, redundant frame reduction and the final video classification. For extracting video features, CNN (GoogleNet or ResNet) is adopted due to its successful applications in image feature extraction. For selecting salient video features, this paper integrates an LSTM into the spatial transformer network to address the problem that the traditional attention mechanisms including soft attention and hard attention cannot utilize the motion information during the attention procedure. For reducing redundant frames, we use the module of temporal coherence analysis to save computation and filter useless information. Finally, convolutional LSTM is adopted for video classification, since it can capture the temporal and spatial information from the salient video features. In this framework, attention and convolutional LSTM modules are incorporated to model the spatial and temporal information of videos. The proposed framework is shown in Fig. 1. First, we extract the feature maps of each frame from a video by feeding them to a pre-trained GoogleNet. Second, the LSTM spatial transformer network automatically selects the discriminative parts of the feature maps. Third, the transformed feature maps are fed to a convolutional LSTM (ConvLSTM) module to make the prediction for the corresponding frame. Fourth, the predictions of all the frames determine the final classification together. In the following parts in this section, we present the process and the loss function of the model in details. Further, we give the analysis of temporal coherence for reducing video redundancy.

3.1 Convolutional features

To utilize the deep spatial features extracted by CNN, the GoogleNet pre-trained on ImageNet dataset is fed with the frame to generate the feature maps of the last convolutional layer. Feature maps can be viewed as a box of $H \times W \times D$, $U \in R^{H \times W \times D}$, where H , W , and D indicate the height, width, and channel sizes respectively ($7 \times 7 \times 1024$ in our experiment). Moreover, the box can be divided into $H \times W$ D -dimension feature slices, each of

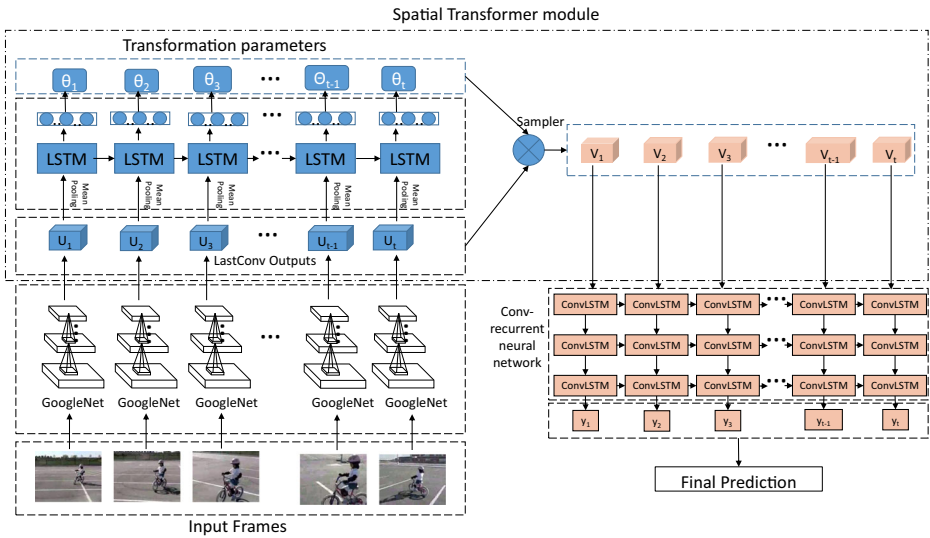


Fig. 1 The framework of convolutional recurrent neural network with attention mechanism for video action recognition

which is a representation corresponding to a spatial locality of the frame. The coordinates of those spatial feature slices can be sampled automatically by the following LSTM spatial transformer network.

3.2 Attention mechanism

The spatial transformer network is an effective module applied in image recognition [13], which consists of three parts: the localization network, the grid generator and the sampler. We modify the original network by embedding an LSTM module to apply it on video recognition. The spatial transformer network is adopted to extract the salient features of videos. The LSTM localization network produces the affine transformation parameter for subsequent spatial transformation. The grid generator uses the affine transformation parameters to compute the mapping of the coordinates between the source and target vector spaces. According to the coordinate correspondence, the sampler generates the value at each target coordinate by bilinear sampling kernel.

3.2.1 Localization network

The localization network can be formulated as $\theta = f_{loc}(U)$, where $U \in R^{H \times W \times D}$ denotes the input feature box extracted by GoogleNet, $f_{loc}(U)$ denotes a map from $R^{H \times W \times D}$ to $R^{2 \times 3}$, and $\theta \in R^{2 \times 3}$ denotes the parameter for affine transformation. Different from a single image, a frame is related to not only itself, but also its neighbor frames. Therefore, an LSTM module is used to model the temporal information in order to compute θ for each frame in a video. Figure 2 shows the process of mapping the features into affine transformation parameters by the LSTM localization network. First, CNN converts each frame in the video into a feature box with the size of $H \times W \times D$ (here $7 \times 7 \times 1024$). Second, each feature box is reduced in spatial dimensions (H and W) and cast into a D-dimensional

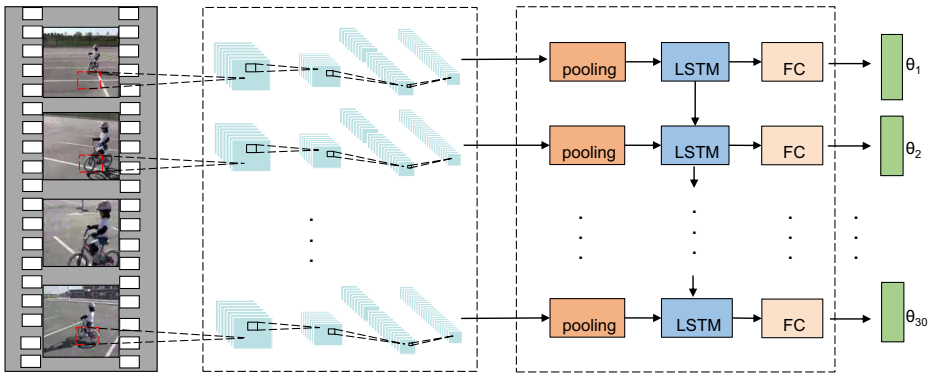


Fig. 2 LSTM localization network. For instance, the feature maps extracted from a video clip of 30 frames are pooled by using an average pooling layer and are then integrated in an LSTM cell. The following fully connected layer receives the hidden states θ of the LSTM cell to compute transformation parameters θ_i for each frame

feature vector through the mean pooling layer, which brings less computation consumption. Third, feature vectors are input into the LSTM by order of time to produce the hidden state at each moment. Fourth, each hidden state is input into the fully connected layer with a linear activation functions to obtain the affine transformation parameter θ for each frame. The subsequent spatial transformation is essentially the mapping of 2-dimensional coordinates between two vector spaces, i.e., affine transformation. Affine transformation is implemented by linear and translation transformations. The parameters of linear and translation transformations are the matrices with the size of 2×2 and 2×1 , respectively. Therefore, the parameter for affine transformation is denoted as $R^{2 \times 3}$.

3.2.2 Grid generator

The transformation parameter θ is used to compute the sampling coordinates by 2D affine transformation. Let $V \in R^{H' \times W' \times D}$ denote the output feature box (the transformed feature maps), where H' , W' , and D denote the height, width and channel sizes respectively. The coordinate of the i -th feature slice in the output feature box is denoted by (x_i^t, y_i^t) . All of them are stored in $G = \{G_i\}$, where $G_i = (x_i^t, y_i^t)$. The grid generator is implemented using affine transformation $T_\theta(\cdot)$, which is formulated as:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = T_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1}$$

where (x_i^s, y_i^s) denotes the coordinate of the feature slice in the input feature box to be sampled, A_θ denotes the affine transformation matrix $\begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{pmatrix}$. Practically, we normalize the height and width to $[-1, 1]$.

3.2.3 The sampler

To perform spatial transformation, the sampler computes each value in the target feature box V by applying a sampling kernel centered at a particular coordinate from the input feature

box U . The sampling coordinates can be deduced by 2D affine transformation parameterized by θ . As shown in Fig. 3, the sampler adopts the Bilinear Sampling Kernel [13] to get the output feature box V from the input feature box U . The value at the c -th channel of the i -th feature slice in V is computed by:

$$V_i^c = \sum_n^H \sum_m^W U_{mn}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \tag{2}$$

The derivative of V_i^c with respect to U_{nm}^c and A_θ are computed as follows:

$$\frac{\partial V_i^c}{\partial U_{nm}^c} = \sum_n^H \sum_m^W \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \tag{3}$$

$$\begin{aligned} \frac{\partial V_i^c}{\partial x_i^s} &= \sum_n^H \sum_m^W U_{mn}^c \max(0, 1 - |y_i^s - n|) \\ &= \begin{cases} 0, & |m - x_i^s| \geq 1 \\ 1, & m \geq x_i^s \\ -1, & m < x_i^s \end{cases} \end{aligned} \tag{4}$$

$$\begin{aligned} \frac{\partial V_i^c}{\partial y_i^s} &= \sum_n^H \sum_m^W U_{mn}^c \max(0, 1 - |x_i^s - m|) \\ &= \begin{cases} 0, & |m - y_i^s| \geq 1 \\ 1, & m \geq y_i^s \\ -1, & m < y_i^s \end{cases} \end{aligned} \tag{5}$$

$$\frac{\partial x_i^s}{\partial \theta_{11}} = x_i^t, \quad \frac{\partial x_i^s}{\partial \theta_{12}} = y_i^t, \quad \frac{\partial x_i^s}{\partial \theta_{13}} = 1 \tag{6}$$

$$\frac{\partial x_i^s}{\partial \theta_{21}} = 0, \quad \frac{\partial x_i^s}{\partial \theta_{22}} = 0, \quad \frac{\partial x_i^s}{\partial \theta_{23}} = 0 \tag{7}$$

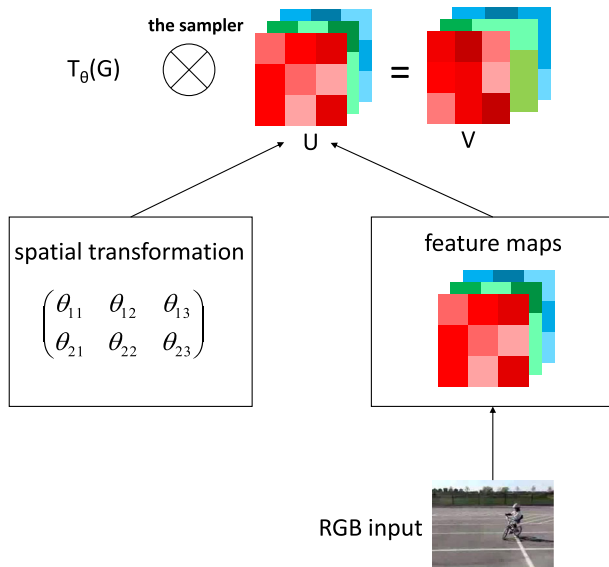


Fig. 3 An example of transforming feature maps. After computing θ , the transformed feature maps V are constructed by sampling the feature maps U at multiple coordinates. The sampling coordinates can be deduced by 2D affine transformation parameterized by θ

$$\frac{\partial y_i^s}{\partial \theta_{11}} = 0, \frac{\partial y_i^s}{\partial \theta_{12}} = 0, \frac{\partial y_i^s}{\partial \theta_{13}} = 0 \tag{8}$$

$$\frac{\partial y_i^s}{\partial \theta_{21}} = x_i^t, \frac{\partial y_i^s}{\partial \theta_{22}} = y_i^t, \frac{\partial y_i^s}{\partial \theta_{23}} = 1 \tag{9}$$

It is obvious that the derivatives can be propagated to both the GoogleNet and the LSTM spatial transformer network.

3.3 Convolutional LSTM

The traditional LSTM is explicitly designed to learn long-term dependencies, but its performance deteriorates when processing videos containing not only temporal information but also spatial information. Because LSTM cannot model spatial information well, it is necessary to convert the spatiotemporal data to the temporal data via a pooling or fully connected layer, i. e., the spatial contents are abandoned. The convolutional LSTM overcomes that problem by substituting convolution for multiplication [23]. As shown in Fig. 4, the class of each frame can be predicted sequentially by inputting feature maps to the convolutional LSTM module. The following formulas define the details of the convolutional LSTM:

$$i^{(t)} = \sigma(W_{xi} * x^{(t)} + W_{hi} * h^{(t-1)} + W_{ci} \circ c^{(t-1)} + b_i) \tag{10}$$

$$f^{(t)} = \sigma(W_{xf} * x^{(t)} + W_{hf} * h^{(t-1)} + W_{cf} \circ c^{(t-1)} + b_f) \tag{11}$$

$$o^{(t)} = \sigma(W_{xo} * x^{(t)} + W_{ho} * h^{(t-1)} + W_{co} \circ c^{(t-1)} + b_o) \tag{12}$$

$$G^{(t)} = \tanh(W_{xc} * x^{(t)} + W_{hc} * h^{(t-1)} + b_c) \tag{13}$$

$$c^{(t)} = f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ G^{(t)} \tag{14}$$

$$h^{(t)} = o^{(t)} \circ \tanh(c^{(t)}) \tag{15}$$

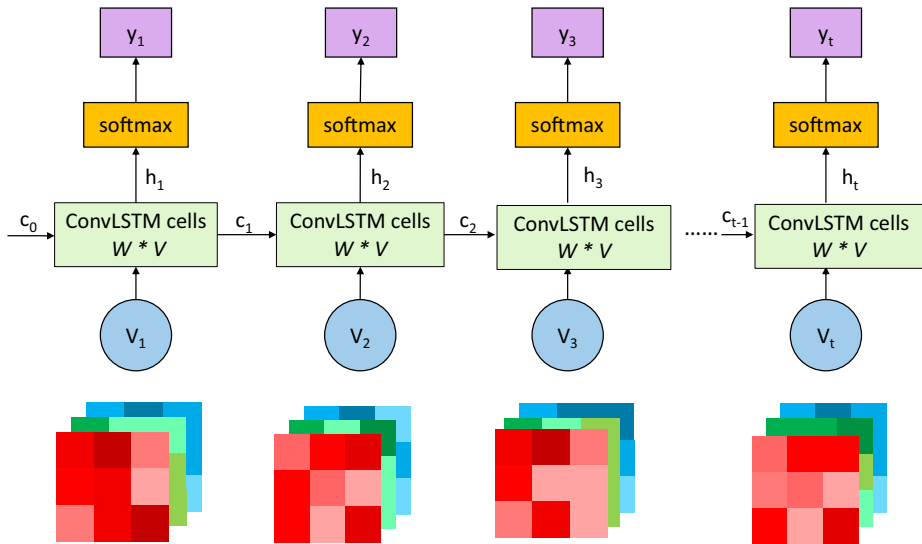


Fig. 4 Convolutional LSTM for prediction of each frame. At the t -th time, Convolutional LSTM receives the feature maps of the t -th frame and integrates them in the history memory. Instead of memorizing temporal information only, the spatial and temporal information from time 0 to t captured by convolutional LSTM are utilized by a softmax layer to predict the class of the t -th frame with higher confidence

where $*$ denotes convolution, \circ denotes element wise multiplication, $W_{x\sim}$ and $W_{h\sim}$ represent 2D convolutional kernels, b_{\sim} is bias vectors, and σ denotes the sigmoid function. The input, forget and output gates $i^{(t)}$, $f^{(t)}$ and $o^{(t)}$, the candidate memory $G^{(t)}$, memory cells $c^{(t)}$ and $c^{(t-1)}$, and hidden states $h^{(t)}$ and $h^{(t-1)}$ are 3D tensors. The convolutional LSTM can receive the feature maps of each frame while keeping the spatial dimensions invariable. Intuitively, stacking several convolutional LSTMs forms a more powerful deep architecture.

3.4 Initialization strategy and loss function

In order to accelerate the convergence of the LSTM module in the LSTM spatial transformer, the same strategy as in [32] is used to compute the initial memory cell $c^{(0)}$ and hidden state $h^{(0)}$:

$$c^{(0)} = f_{init,c} \left(\frac{1}{T} \sum_{t=1}^T \left(\frac{1}{H \times W} \sum_{i=1}^{H \times W} x_{t,i} \right) \right) \tag{16}$$

$$h^{(0)} = f_{init,h} \left(\frac{1}{T} \sum_{t=1}^T \left(\frac{1}{H \times W} \sum_{i=1}^{H \times W} x_{t,i} \right) \right) \tag{17}$$

where $f_{init,c}$ and $f_{init,h}$ indicate two multilayer perceptrons, T denotes the frame number in a video, H and W denote the height and width of the input feature box respectively, and $x_{t,i}$ denotes the i -th feature slice.

The loss function is set up based on cross entropy and double random penalty. It is formulated as follows:

$$l = - \sum_{t=1}^T \sum_{i=1}^C y_{t,i} \log \hat{y}_{t,i} + \gamma \sum_i \sum_j \theta_{i,j}^2 \tag{18}$$

where y_t denotes the one hot label vector, $\hat{y}_{t,i}$ denotes the probability of the i -th class predicted at the time t , T and C denote the numbers of time step and class respectively, γ denotes the weight decay coefficient and θ denotes the parameters of our model.

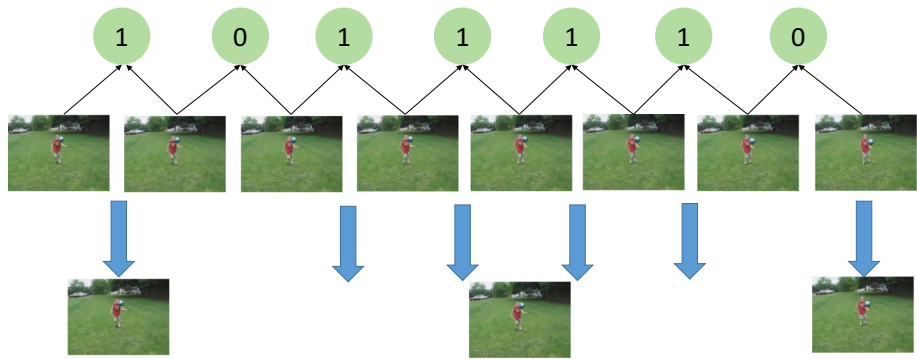
3.5 The analysis of temporal coherence

Using vector representations of videos, the analysis of temporal coherence can be employed to reduce redundant video frames effectively. Different from using handcrafted features as in [31], We use GoogleNet to extract the D' -dimensional vector representations p_i for each frame, $P = [p_1, p_2, \dots, p_i, \dots, p_n]$, where n denotes the number of frames. The correlation between each pair of consecutive frames, $C = [C_1, C_2, \dots, C_{n-1}]$, is calculated according to:

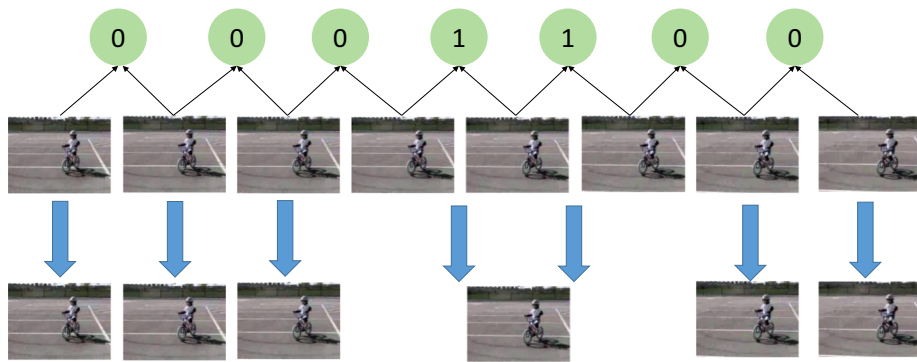
$$C_t = \sqrt{\sum_{d'=1}^{D'} (p_{t,d'} - p_{t+1,d'})^2} \tag{19}$$

$$C'_t = \begin{cases} 0, & C_t < c_{t,k} \\ 1, & C_t \geq c_{t,k} \end{cases} \tag{20}$$

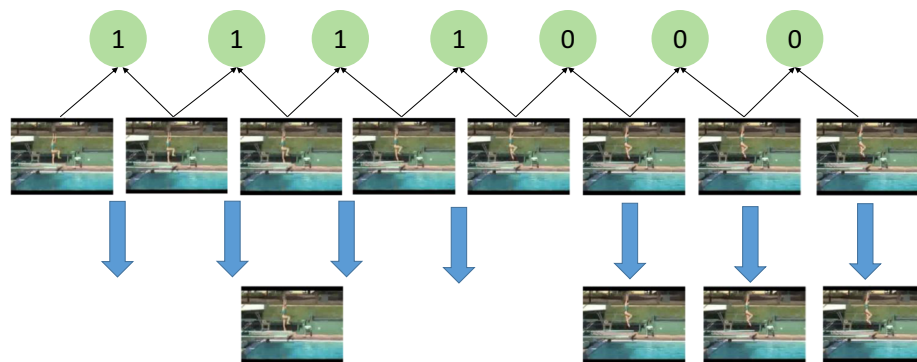
A video is a sequence of temporally contiguous frames. Subsequences are the temporally contiguous video clips split from the whole video sequence at some key points, where the similarity of two adjacent frames is less than a certain threshold $c_{t,k}$. The threshold $c_{t,k}$ denotes the mean value of the correlation vectors in C . $C'_t = 1$ indicates that the t -th and $(t + 1)$ -th frames belong to the same subsequence. $C'_t = 0$ indicates that the two frames



(a) Coherence analysis of juggling



(b) Coherence analysis of biking



(c) Coherence analysis of diving

Fig. 5 Three examples of reserving frames for each subsequence. In part **a**, 8 frames from a short clip are grouped into 3 subsequences by computing the Euclidian distances between their vector representations extracted by GoogleNet, where 1 indicates the neighbor frames are correlated and 0 indicates they are uncorrelated. In part **b**, the frames are grouped into three subsequences and the frames in part **c** are grouped into two subsequences

do not belong to the same subsequence. We keep just one frame for each subsequence, so that the redundant feature maps can be reduced. Figure 5 shows an example of reducing redundant frames. Furthermore, we incorporate the analysis of temporal coherence to reduce redundant feature maps extracted by GoogleNet, as shown in Fig. 6. In this way, before being fed to the LSTM spatial transformer network, the redundant feature maps extracted by GoogleNet are reduced. For example, since U_1 and U_2 belong to the same subsequence, either of them can be reduced and the other reserved. However, U_3 constitutes a subsequence itself, so no frames can be reduced in this case. Those feature maps are then converted to vectors and used in the LSTM spatial transformer for latter sampling. After sampling, the convolutional LSTM module following by a softmax layer is used to predict the class for each frame. The class of the video is finally generated by voting.

4 Experiment

4.1 Datasets

To evaluate the effectiveness of the proposed method, experiments are conducted on three datasets of realistic actions: UCF-11 (YouTube Action Data Set) [24], HMDB-51 (a large human motion database) [18], and UCF-101 (an extension of UCF-11) [35]. UCF-11 contains 1600 videos, each of which belongs to one of the 11 action categories: basketball shooting, biking, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking and walking with a dog. In our experiments, 975 videos are used for training and the rest for testing. HMDB-51 provides three splits, each of which contains 5100 videos, including the 3570 for training and the rest for testing. All the videos cover 51 action categories including clapping, drinking, hugging, jumping, somersaulting, throwing, etc. UCF-101 contains 13320 videos from 101 action categories, and it is one of the most challenging datasets to date. Since CNN requires inputs of fixed size, all the frames in a video are cropped to 224 times 224 and sequentially fed to a GoogleNet pre-trained on the ImageNet dataset. The output of the last convolution layer is adopted as the high-level representation for each frame.

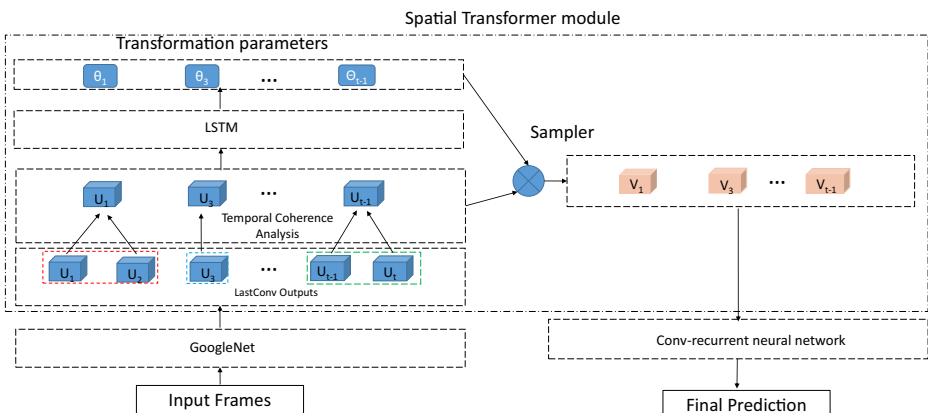


Fig. 6 The framework of ConvLSTM with attention mechanism using the analysis of temporal coherence for reducing redundant feature maps

Table 1 Hyper parameters

Parameter	Value
Learning rate	10^{-4}
The size of the convolution kernel	3×3
The number of the convolution kernel	16
Dropout rate	0.5
Weight decay coefficient	10^{-5}

4.2 Experimental settings

Adam optimization algorithm [16] is used to optimize the loss function. The architecture of the convolutional LSTM module is selected by cross validation. Other hyper parameters such as learning rate, dropout rate and weight decay coefficient are also selected by cross validation. The values of those hyper parameters are listed in Table 1.

According to the settings in Table 1, five recurrent convolutional models with 1 up to 5 stacked recurrent layers are evaluated on UCF-11, HMDB-51 and UCF-101, respectively. The results of them are shown in Table 2. It can be seen that the 3-layer stacked convolutional LSTM achieves the best results on UCF-11 and HMDB-51. The 3-layer and 4-layer stacked convolutional LSTMs achieve almost the same results on UCF-101. Balancing the accuracy and efficiency, the 3-layer stacked convolution LSTM is used in the following experiments. Our model is trained by using a Tesla P40 GPU.

The differences among hardware and software environments will directly affect the time performance of the model. Therefore, to evaluate the scalability of the model objectively, space and time complexity is deduced here. The number of parameters included in the weights (convolutional kernels) is used to represent space complexity, and the number of floating-point multiplication operations involved is used to represent time complexity. The three parameterized parts of the model are the CNN module, the spatial transformer module, and the stacked convolutional LSTM module. A video frame of $224 \times 224 \times 3$ is input into the CNN module (for example, GoogleNet) to output a feature box with the size of $7 \times 7 \times 1024$, wherein the number of parameters included in convolutional kernels is about 5.86M. The number of floating-point multiplication operations is approximately 1.58T. The spatial transformer module consists of an LSTM (256 hidden units) and a fully connected layer (6 neurons), so there are 1.31M parameters as well as 1.31M floating-point multiplication operations. The stacked convolutional LSTM module contains 1 ~ 5 Conv-LSTMs, each of which outputs the $7 \times 7 \times 256$ feature box, with 3×3 kernels. Except that the first Conv-LSTM maps 1024 channels to 128, other Conv-LSTMs keep the number of channels as 128 unchanged. The number of parameters of the stacked Conv-LSTMs is

Table 2 Accuracy of multi-layer ConvLSTM on different datasets

Model	UCF-11	HMDB-51	UCF-101
1-ConvLSTM	93.27	63.54	89.23
2-ConvLSTM	93.78	65.95	90.17
3-ConvLSTM	94.12	67.12	92.51
4-ConvLSTM	94.05	67.08	92.54
5-ConvLSTM	93.84	66.37	92.39

Table 3 Recognition accuracy on UCF-11 using different methods

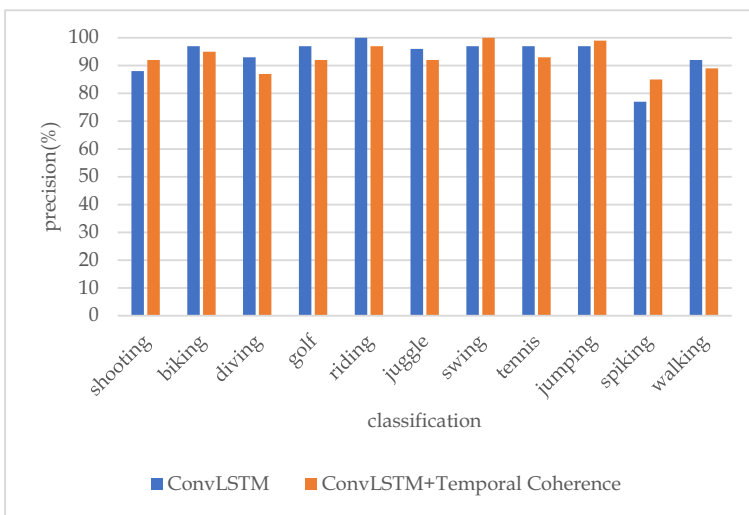
Model	UCF-11
Multiple feature + Multiple instance learning [12]	75.21
Stacked convolutional ISA [20]	75.8
Bag-of-visual words (BoW) [2]	76.5
Dense trajectories + Motion boundary descriptors [45]	84.9
Soft attention model (@ 30 fps, = 0) [32]	84.96
Baseline	83.81
Our model	94.12
Our model + the analysis of temporal coherence	93.48

$5.31M + (n - 1) \times 1.18M$ (n -layer Conv-LSTMs). The number of floating-point multiplication operations is $260.11M + (n - 1) \times 57.8M$. It can be concluded from the above analyses that the scalability of the model depends mainly on the CNN module and linearly increases as the number of stacked Conv-LSTMs increases.

In the experiments, the frame rate of the videos used in this paper is 30 fps. A video more than 1 second is sliced into multiple clips, each of which contains 30 frames. During the inference stage, the category of a frame is directly predicted by the convolutional LSTM module and then the category of a clip is generated by its frames via majority voting. Similarly, the category of the entire video is voted by its clips.

4.3 Results on UCF-11

The proposed method is evaluated on the relatively small dataset UCF-11. Results obtained by our method on UCF-11 and results of some current state-of-the-art methods are listed in Table 3. As we can observe, the proposed method outperforms the listed models by nearly 10%, which demonstrates the effectiveness of the proposed algorithm.

**Fig. 7** Comparison for influence on performance on UCF-11 of the temporal coherence for different actions

To clarify the effectiveness of the spatial transformer network, we remove the spatial transformer network and keep the basic architecture as our baseline. The result obtained by the baseline model is also presented in Table 3. By comparing the performances of the models with and without the spatial transformer network, it can be seen that the spatial transformer network obviously improves the performance.

To verify the effectiveness of temporal coherence analysis, the model with temporal coherence analysis is tested on UCF-11 and the result is presented in Table 3. Further, the detailed results of each class are shown in Fig. 7. With the analysis of temporal coherence, our model is trained 30% faster, with overall accuracy loss less than 1%. Overall, the temporal coherence saves considerable time by reducing redundant frames with a small loss of accuracy. Figure 7 shows that the classification accuracy decreases for most categories and increases for others. This is because the proportion of redundant frames varies with the categories of videos. For example, the “walking” action spans across the entire video. So the proportion of redundant frames in such durative actions is low. Useful information is easily ignored in the process of temporal coherence, which leads to the decrease of classification accuracy. However, the actions of “spiking”, “shooting” and “jumping” are contained in a few key frames at the moment when actions happen, so the proportion of redundant frames in such non-durative actions is high. Useless information can be filtered through temporal coherence, thereby improving the classification accuracy.

4.4 Results on HMDB-51

The proposed method is also evaluated on the big dataset HMDB-51 and is compared with the results of state-of-the-art methods in Table 4. The methods in the top part only take original RGB data as inputs. Those in the bottom not only make use of RGB data but also optical flow data. From Table 4, we can see our proposed method obtains competitive results when using RGB and optical flow data. However, the composite LSTM achieves the best results on HMDB-51. The main reasons lie in the following two folds: 1) After pre-trained on ImageNet, the composite LSTM is further pre-trained on 300 hours of YouTube videos and finally fine-tuned on video classification datasets. In comparison, our model is only pre-trained on ImageNet. 2) The composite LSTM is constructed under the framework

Table 4 Recognition accuracy on HMDB-51 using different methods

Model	HMDB-51
Spatial stream ConvNet [33]	40.5
Soft attention model [32]	41.3
Composite LSTM Model [36]	44.0
Baseline	40.8
Our model	42.2
FST CN [37]	59.1
Two-stream ConvNet [33]	59.4
Video Darwin [7]	63.7
Multi-skip Feature Stacking [19]	65.1
Two-stream Fusion [6]	65.4
Fisher Vectors + Stacked Fisher Vectors [28]	66.8
Baseline + Optical flow	64.1
Our model + Optical flow	67.1

of auto-encoder by utilizing the complex mechanism, which combines reconstructing the input and predicting the future in the pre-training phase. Figure 8 represents the detailed results for each action in HMDB-51. The diverse classification accuracy may be caused by different video qualities. By comparing the results obtained by the models with and without

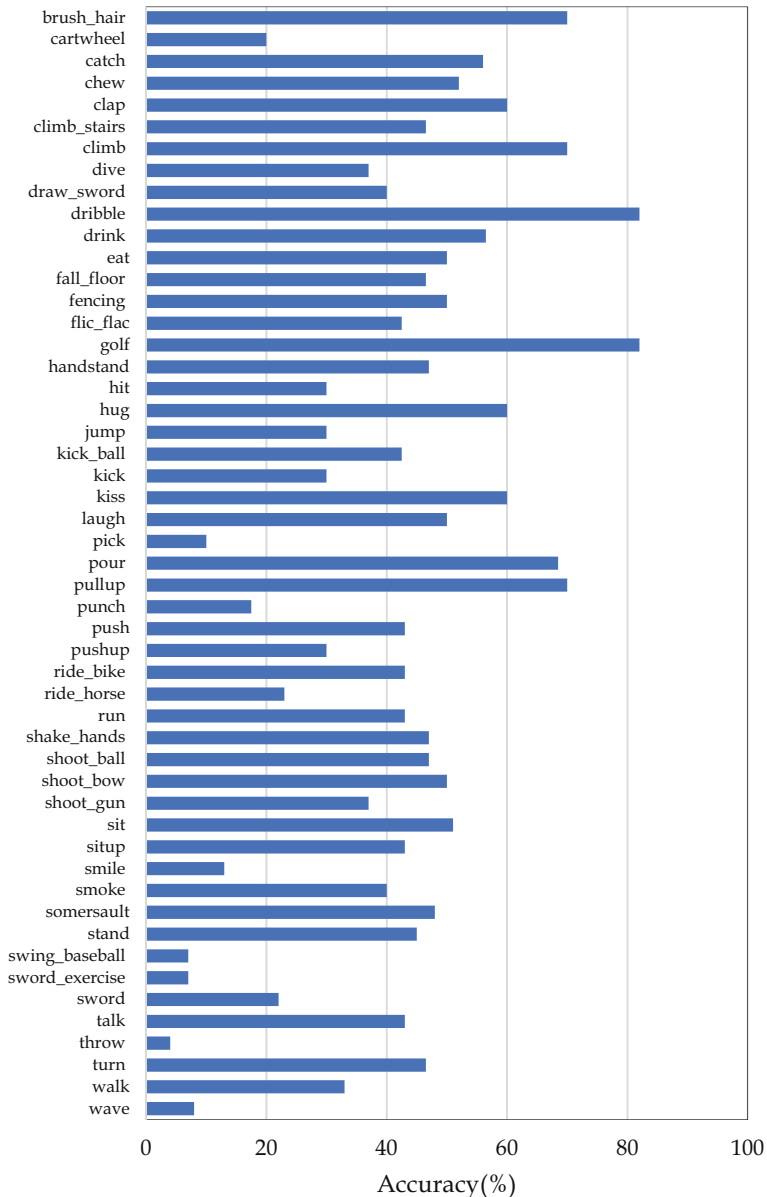


Fig. 8 Recognition accuracy on HMDB-51 for different actions (using only RGB data)

Table 5 Recognition accuracy on UCF-101 using different methods

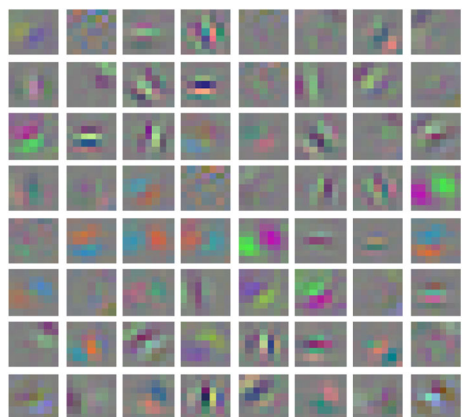
Model	UCF-101
Spatial stream ConvNet [33]	73.0
Composite LSTM Model [36]	75.8
Our model-GoogleNet	76.5
Our model-ResNet	76.7
Two-stream ConvNet [33]	88.0
Composite LSTM Model [36]	84.3
Multi-skip Feature Stacking [19]	89.1
Conv Pooling [27]	88.6
FST CN [37]	88.1
Two-stream Fusion [6]	92.5
P3D [30]	88.6
CoViAR [50]	94.9
Our model-GoogleNet + Optical flow	92.5
Our model-ResNet + Optical flow	92.8

the spatial transformer network, it indicates that the spatial transformer network can improve the performance.

4.5 Results on UCF-101

To further validate the effectiveness of the proposed method, the experiment on a larger dataset UCF101 is performed. Here GoogleNet and ResNet are employed as two different feature extractors to observe their impact on the final classification results. Table 5 is composed by two parts, where the top part uses RGB data, the bottom part uses RGB and optical flow data. It can be seen from Table 5 that the results obtained by GoogleNet and ResNet are not significantly different. Using RGB and optical flow data to train the proposed model

Fig. 9 Kernel visualization. The kernels of the first convolutional layer are visualized



can obtain better results than using only RGB data. And the proposed method can achieve results comparable to the state-of-the-art results listed here.

As mentioned in reference [50], compressed videos provide temporal information while avoiding cumbersome computations caused by optical flow data. Therefore, we also tried a new type of two-stream data, RGB + compressed videos, to train the model and observe the effectiveness of compressed videos. As a result, using RGB + compressed videos achieves the similar results as that of using RGB + optical flow and significantly higher results than that of using RGB. To conclude, compressed videos might be a promising substitute for optical flow. On the other hand, compared with RGB + optical flow, no noticeable differences made by the attention module are observed, so the results of RGB + compressed videos are not listed here.

4.6 Inspection to feature maps

In this section, the validity of GoogleNet is evaluated by visualizing intermediate filters and feature maps. Figure 9 visualizes some filters of the first convolution layer in GoogleNet. It is obvious that each filter intends to capture definite features of some color or edge. Thus empirically, the extracted features should be effective. Figure 10 illustrates the original images and the feature maps of the first two convolution layers in GoogleNet. In contrast to

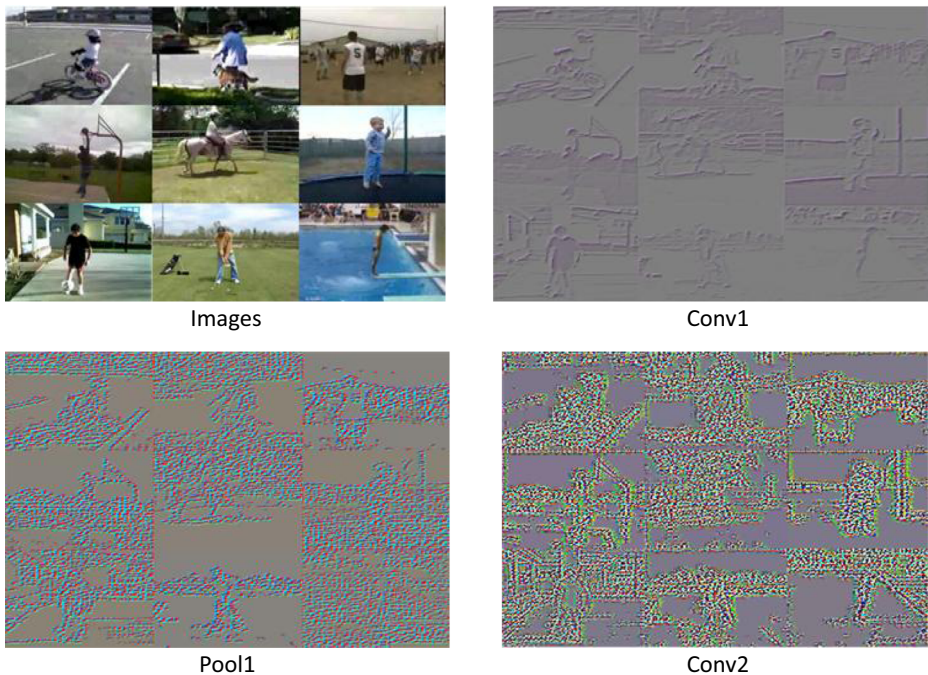


Fig. 10 Feature map visualization. The top left part shows 9 images to be fed to GoogleNet. The top right part illustrates the corresponding feature maps from the first convolutional layer. The feature maps from the first pooling layer and the second convolutional layer are listed in the bottom (from left to right)

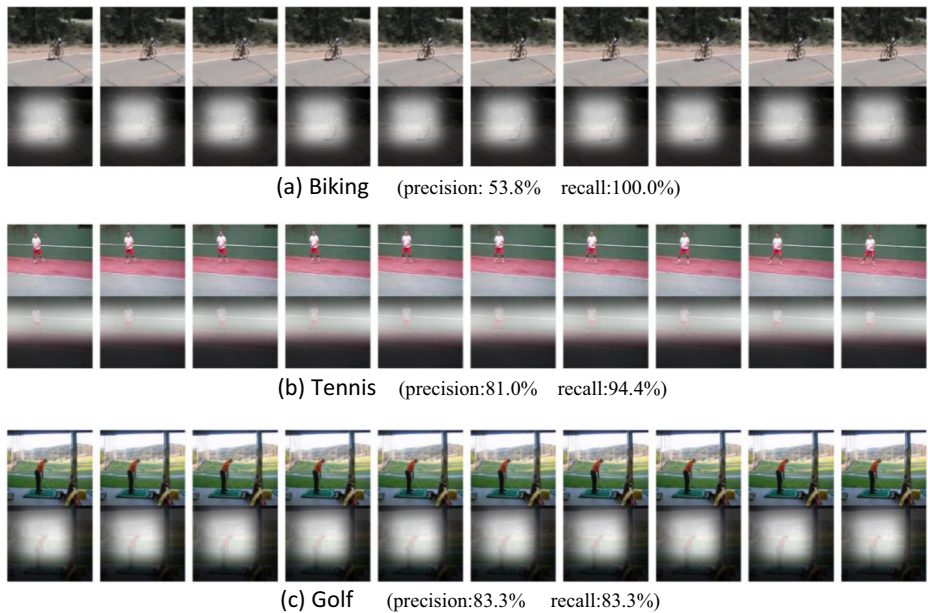


Fig. 11 Correctly classified examples. Three examples, biking, tennis and golf, are shown here. More brighter an area is, more salient the corresponding content is. The regions containing human, the bike, the tennis racket and the golf club are selected as salient information.

the original images, the feature maps indicate some meaningful contents: main objects and shapes. Thus, it is effective of GoogleNet to extract frame features.

4.7 Attention visualization

The attention effects brought by the spatial transformer network are illustrated in Figs. 11 and 12. The brighter a position is, the more heavily it weighs against the corresponding input. Figure 11 shows some correctly classified examples, demonstrating that our attention module is effective for winking out the most discriminative parts.

Some misclassified examples are illustrated in Fig. 12. For Fig. 12a, the proportion of the discriminative patch to the whole frame is too small, making it difficult to extract meaningful content and resulting in misclassification. The proportion may be raised by using the multi-resolution method. For Fig. 12b, another salient object, a red eye-catching car, adds much noise to the walking man. To reduce the noise, we may separate the foreground from the background. For Fig. 12c, the video does not capture the whole human body and thus causes misclassification.

We also make quantitative analyses by using precision and recall. However, the used video classification datasets lack the ground truth of discriminative regions. Therefore, we manually label the discriminative regions for each example in Figs. 11 and 12. Then, the precision and recall are computed and listed under each example. It can be seen that the precision and recall of the discriminative regions for correctly classified examples are obviously higher than that of incorrectly classified examples.

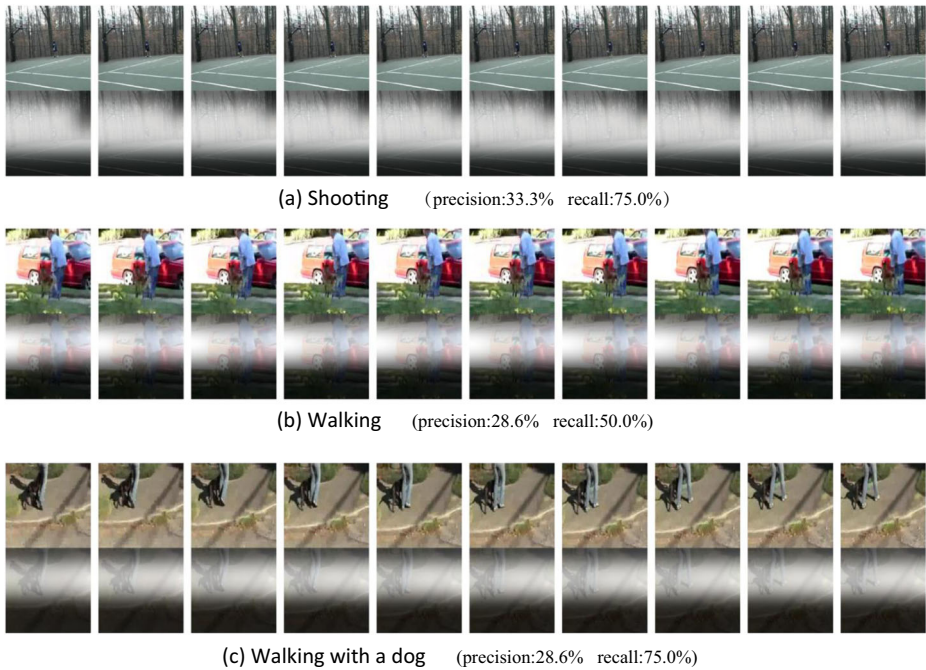


Fig. 12 Incorrectly classified examples. For the example of shooting, the regions containing human are small and difficult to be selected. For the example of walking, a red car appears as noisy background surrounding the walking man. In the last example, only a half of the woman's body is reserved.

5 Conclusion

In order to improve the accuracy of human action recognition, an attention mechanism based convolutional LSTM network is proposed. We incorporate LSTM in the spatial transformer network and adopt the LSTM spatial transformer network to extract the salient regions of the feature maps. The ConvLSTM module is then used to integrate the temporal information of the feature maps with the spatial information sustained. The experiments demonstrate that our model can extract salient feature representations and acquire competitive classification accuracy.

In the future works, because the spatial transformer is based on affine transformation which is not limited to 2D coordinate scenarios, affine transformation may be well applied on 3D coordinate systems, including width, height and time. Also, multi-resolution methods can be used to obtain the fine-grained section and amplify the regions containing humans.

Acknowledgements The authors are grateful to the support of the National Natural Science Foundation of China (61572104, 61103146, 61402076) and the Fundamental Research Funds for the Central Universities (DUT17JC04).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Bahdanau Dzmitry, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: International conference on learning representations ICLR
2. Bhattacharya S, Sukthankar R, Jin R, Shah M (2011) A probabilistic representation for efficient large scale visual recognition tasks. In: IEEE conference on computer vision and pattern recognition, CVPR, vol 42, pp 2593–2600
3. Carreira J, Zisserman A (2017) Quo vadis, action recognition? A new model and the kinetics dataset. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 4724–4733
4. Deng J, Dong W, Socher R, Li LJ, Li K, Li F-F (2009) Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 248–255
5. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 2625–2634
6. Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 1933–1941
7. Fernando B, Gavves E, Oramas MJ, Ghodrati A, Tuytelaars T (2015) Modeling video evolution for action recognition. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 5378–5387
8. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: IEEE international conference on acoustics, speech and signal processing, ICASSP, vol 38, pp 6645–6649
9. Guo Y, Tao D, Liu W, Cheng J (2017) Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition. *IEEE Trans Syst Man Cybern Syst* 47(4):617–627
10. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 770–778
11. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9:1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
12. Ikizler-Cinbis N, Sclaroff S (2010) Object, scene and actions: combining multiple features for human action recognition. In: European conference on computer vision, ECCV, vol 6311, pp 494–507
13. Jaderberg M, Simonyan K, Zisserman A (2015) Spatial transformer networks. In: Advances in neural information processing systems, NIPS, pp 2017–2025
14. Jégou H, Douze M, Schmid C, Pérez P (2010). In: IEEE conference on computer vision and pattern recognition, CVPR, vol 238, pp 3304–3311
15. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Li FF (2014) Large-scale video classification with convolutional neural networks. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 1725–1732
16. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. In: International conference on learning representations ICLR
17. Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, NIPS, pp 1097–1105
18. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: a large video database for human motion recognition. In: IEEE international conference on computer vision, ICCV, vol 24, pp 2556–2563
19. Lan Z, Lin M, Li X, Hauptmann AG, Raj B (2015) Beyond Gaussian pyramid: multi-skip feature stacking for action recognition. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 204–212
20. Le QV, Zou WY, Yeung SY, Ng AY (2011) Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: IEEE conference on computer vision and pattern recognition, CVPR, vol 42, pp 3361–3368
21. Lei Q, Zhang H, Xin M, Cai Y (2018) A hierarchical representation for human action recognition in realistic scenes. *Multimed Tools Appl*, MTAP 3:1–21
22. Li Q, Qiu Z, Yao T, Mei T, Rui Y, Luo J (2016) Action recognition by learning deep multi-granular spatio-temporal video representation. In: ACM on international conference on multimedia retrieval, ICMR, pp 159–166
23. Li Z, Gavves E, Jain M, Snoek CGM (2018) Videolstm convolves, attends and flows for action recognition. *Comput Vis Image Underst* 166:41–50
24. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos. In: IEEE conference on computer vision and pattern recognition, CVPR, vol 38, pp 1996–2003
25. Luo Y, Yin D, Wang A, Wu W (2018) Pedestrian tracking in surveillance video based on modified CNN. *Multimed Tools Appl*, MTAP 77(18):24041–24058

26. Mnih V, Heess N, Graves A (2014) Recurrent models of visual attention. In: Advances in neural information processing systems, NIPS, pp 2204–2212
27. Ng JYH, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: deep networks for video classification. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 4694–4702
28. Peng X, Zou C, Qiao Y, Peng Q (2014) Action recognition with stacked fisher vectors. In: European conference on computer vision, ECCV, vol 8693, pp 581–595
29. Peng X, Wang L, Wang X, Yu Q (2016) Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. *Comput Vis Image Underst* 150:109–125
30. Qiu Z, Yao T, Mei T (2017) Learning spatio-temporal representation with pseudo-3d residual networks. In: IEEE international conference on computer vision, ICCV, pp 5534–5542
31. Saleh A, Abdel-Nasser M, Akram F, Garcia MA, Puig D (2016) Analysis of temporal coherence in videos for action recognition. In: International conference on image analysis and recognition ICIAR
32. Sharma S, Kiros R, Salakhutdinov R (2016) Action recognition using visual attention. In: International conference on learning representations, ICLR
33. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. *Adv Neural Inf Proces Syst, NIPS* 1(4):568–576
34. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, ICLR
35. Soomro K, Zamir AR, Shah M (2012) UCF101: a dataset of 101 human actions classes from videos in the wild, Technical report CRCV-TR-12-01 UCF center for research in computer vision
36. Srivastava N, Mansimov E, Salakhutdinov R (2015) Unsupervised Learning of Video Representations Using LSTMs. In: International conference on machine learning, ICML, pp 843–852
37. Sun L, Jia K, Yeung DY, Shi BE (2015) Human action recognition using factorized spatio-temporal convolutional networks. In: IEEE international conference on computer vision, CVPR, pp 4597–4605
38. Szegedy C et al (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition, CVPR
39. Tao D, Wen Y, Hong R (2016) Multicolumn bidirectional long short-term memory for mobile devices-based human activity recognition. *IEEE Internet Things J* 3(6):1124–1134
40. Tao D, Guo Y, Li Y, Gao X (2018) Tensor rank preserving discriminant analysis for facial recognition. *IEEE Trans Image Process* 27(1):325–334
41. Tran D, Bourdev LD, Fergus R, Torresani L, Paluri M (2014) C3D: generic features for video analysis. *Commun Res Rep, CoRR* 2(7):8
42. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2017) A closer look at spatiotemporal convolutions for action recognition. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 6450–6459
43. Veeriah V, Zhuang N, Qi GJ, Differential recurrent neural networks for action recognition (2015). In: IEEE international conference on computer vision, CVPR, pp 4041–4049
44. Wang H, Schmid C (2013) Action recognition with improved trajectories. In: IEEE international conference on computer vision, CVPR, pp 3551–3558
45. Wang H, Klaser A, Schmid C, Liu CL (2013) Dense trajectories and motion boundary descriptors for action recognition. *Int J Comput Vis* 103(1):60–79
46. Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep-convolutional descriptors. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 4305–4314
47. Wang X, Farhadi A, Gupta A (2016) Actions transformations. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 2658–2667
48. Wu Z, Wang X, Jiang YG, Ye H, Xue X (2015) Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In: ACM International Conference on Multimedia, pp 461–470
49. Wu Z, Jiang YG, Wang X, Ye H, Xue X (2016) Multi-stream multi-class fusion of deep networks for video classification. In: ACM Conference on Multimedia, pp 791–800
50. Wu CY, Zaheer M, Hu H, Manmatha R, Smola AJ, Krähenbühl P (2018) Compressed video action recognition. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 6026–6035
51. Xu K, Ba J, Kiros R et al (2015) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning, ICML, pp 2048–2057
52. Xu W, Xu W, Yang M, Yu K (2012) 3D convolutional neural networks for human action recognition, vol 35, pp 221–231
53. Yan Y, Ni B, Yang X (2017) Predicting human interaction via relative attention model. In: International joint conference on artificial intelligence, IJCAI, pp 3245–3251
54. Yao L, Torabi A, Cho K, Ballas N, Pal C, Larochelle H, Courville A (2015) Describing videos by exploiting temporal structure. In: IEEE international conference on computer vision, ICCV, pp 4507–4515

55. Ye H, Wu Z, Zhao RW, Wang X, Jiang YG, Xue X (2015) Evaluating two-stream CNN for video classification. In: ACM international conference on multimedia retrieval, ICMR, pp 435–442
56. Zhu Y, Zhao C, Gun H, Wang J, Zhao X, Lu H (2019) Attention CoupleNet: fully convolutional attention coupling network for object detection. *IEEE Trans Image Process* 28(1):113–126



Hongwei Ge received B.S. and M.S. degrees in mathematics from Jilin University, China, and the Ph.D. degree in computer application technology from Jilin University, in 2006. He is currently a professor and a vice dean in the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests are machine learning, computational intelligence, optimization and modeling, computer vision, deep learning. He has published more than 50 papers in these areas. His research was featured in the *IEEE Transactions on Cybernetics*, *IEEE Transactions on Evolutionary Computation*, *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, *Pattern Recognition*, *Information Science*, etc.



Zehang Yan received B.S. and M.S. degrees from Dalian University of Technology, China, in 2012 and 2015 respectively. He is currently a Ph.D student in the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His main research interests lie in machine learning and computer vision.



Wenhao Yu received the Ph.D degree from Dalian University of Technology, China, in 2016. He is currently a software developer in Qunar, Beijing, China.



Liang Sun received the B.E. degree in computer science and technology from Xidian University, Xi'an, China, and the M.S. degree in computer application technology from Jilin University, Changchun, China, in 2003 and 2006, respectively. During 2006?2009, as a D.E. candidate, he was at College of Computer Science and Technology, Jilin University China. During 2009?2012, as a D.E. candidate, he was at Kochi University of Technology (KUT), Japan, as an international student of cooperation between KUT and Jilin University. He received double Dr. degree from Kochi University and Jilin University in March, 2012 and June 2012, respectively. He is currently with the College of Computer Science and Technology, Dalian university of technology, Dalian, China. His main research interests lie in machine learning, deep learning and computer vision.