Pattern Recognition 92 (2019) 13–24

Contents lists available at ScienceDirect

Pattern Recognition

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

Spatiotemporal distilled dense-Connectivity network for video action recognition

Wangli Hao[a,c], Zhaoxiang Zhang[a,b,c,*]

[a] Center of Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA) Beijing, 100190, China
[b] Center for Excellence in Brain Science and Intelligence Technology (CEBSIT) Beijing, 100190, China
[c] University of Chinese Academy of Sciences (UCAS) Beijing, 100190, China

ARTICLE INFO

Article history:
Received 13 August 2018
Revised 16 January 2019
Accepted 2 March 2019
Available online 9 March 2019

Keywords:
Two-stream
Action recognition
Dense-connectivity
Knowledge distillation

ABSTRACT

Two-stream convolutional neural networks show great promise for action recognition tasks. However, most two-stream based approaches train the appearance and motion subnetworks independently, which may lead to the decline in performance due to the lack of interactions among two streams. To overcome this limitation, we propose a Spatiotemporal Distilled Dense-Connectivity Network (STDDCN) for video action recognition. This network implements both knowledge distillation and dense-connectivity (adapted from DenseNet). Using this STDDCN architecture, we aim to explore interaction strategies between appearance and motion streams along different hierarchies. Specifically, block-level dense connections between appearance and motion pathways enable spatiotemporal interaction at the feature representation layers. Moreover, knowledge distillation among two streams (each treated as a student) and their last fusion (treated as teacher) allows both streams to interact at the high level layers. The special architecture of STDDCN allows it to gradually obtain effective hierarchical spatiotemporal features. Moreover, it can be trained end-to-end. Finally, numerous ablation studies validate the effectiveness and generalization of our model on two benchmark datasets, including UCF101 and HMDB51. Simultaneously, our model achieves promising performances.

1. Introduction

Video-based action recognition is an intensively researched field in computer vision, with many approaches progressively proposed that focus on aspects of hand-crafted representations [1–5] and deeply-learned representations [6–15]. Recent developed two-stream based methods further promote the action recognition performance to a new record [7,16–19].

However, most conventional two-stream based action recognition methods train the appearance and motion streams entirely independently and there are no interactions between them except at the last fusion layer. We argue that the lack of interactions between appearance and motion paths yields sub-optimal performance. Although some work has established residual connections among two streams [17], multiscale information has not been leveraged in their design. Thus, we explore the following strategies to build the efficient hierarchical interactions between two streams, in order to get improved action recognition performance.

Dense Convolutional Networks (DenseNet) [20] connect each layer to every other layer in a feed-forward fashion to yield an efficient feature representation model. Specifically for each layer, its input is the feature maps of all preceding layers. In addition, its own feature maps are treated as the inputs for all subsequent layers. Benefiting from its particular structure, DenseNet exhibits several promising benefits, including mitigating the vanishing-gradient problem, strengthening feature propagation, encouraging feature reuse and substantially decreasing the number of parameters.

Our model is partially inspired by the DenseNet and try to generalize dense-connectivity into spatiotemporal domain, via building block-level dense connections between appearance and motion streams. Concretely, the input of the current block of appearance DenseNet is the fusion of two streams. One is the original input from the appearance stream. The other is the feature maps of all preceding blocks from motion stream. Their fusion is realized by the multiplicative gate mechanism. We should note that, the block-level dense connection established here is unidirectional that from

* Corresponding author to: Center for Brain Inspired Intelligence, No. 95 Zhongguancun East Road, Beijing 100190, China.
E-mail address: zhaoxiang.zhang@ia.ac.cn (Z. Zhang).

https://doi.org/10.1016/j.patcog.2019.03.005
0031-3203/

motion stream to appearance stream, which is because the appearance stream dominate the motion stream during training [17]. Block-level densely connected two-stream network permits effective spatiotemporal interaction at the feature representation layers.

Knowledge distillation [21], a new emerging knowledge transfer strategy, which is realized by transferring knowledge learned from the teacher network into the corresponding student. Specifically, the complementary information of network *A* to network *B* can be seen as the knowledge that transferred from *A* to *B*. Concerning action recognition scenario, we believe that the appearance and motion streams contain mutual complementary information. In addition, the fusion of two streams, a new distribution of video data, also carries complementary information to both two streams. Consequently, a new knowledge distillation module is developed, expecting to thoroughly fuse the complementary information from appearance and motion streams. In detail, our knowledge distillation module contains two students and a teacher, with each student gaining complementary knowledge learned from the other student and the teacher. Two students refer to the output of appearance and motion streams respectively. In addition, the teacher refers to the final fusion of two-stream outputs. The proposed knowledge distillation module allows two streams to interact effectively at the high level layers. The key differences between our model and [21] are that, on the one hand, the knowledge distillation performed in our model is between two ConvNets with same architectures (RGB DenseNet and Flow DenseNet) based on different modalities of the same data (RGB modality and Flow modality of the same video respectively). Whereas the distillation executed in [21] is between two networks with different architectures (for example, ResNet18 and ResNet34) based on the same data. Namely, the distilled knowledge in [21] is based on different networks, but in our model is based on different modalities. On the other hand, our model performs mutual distillation between two student networks and knowledge distillation from the fusion of two streams to each of them respectively, whereas the distillation in [21] only from one network (teacher network) to the other (student network). The key contribution on this point is that our distillation can be seen as two students performing mutual learning and additionally learn knowledge from the teacher. These students and the teacher can be seen as cohort and learn collaboratively, all members become somewhat more similar by learning to mimic each other, which will leads to better action recognition performance.

Block-level densely connected two-stream networks coupled with knowledge distillation module formed our final Spatiotemporal Distilled Dense-Connectivity Network (STDDCN) for video action recognition. STDDCN possesses some compelling advantages. For example, STDDCN allows effective interactions among appearance and motion streams at different level layers, which encourages the acquisition of hierarchical complex spatiotemporal features. Moreover, STDDCN can be trained end-to-end.

To validate the performance of STDDCN, extensive ablative experiments were performed based on two benchmark datasets, UCF101 [22] and HMDB51 [23]; and in summary our model obtains promising action recognition results.

Contributions of our paper can be summarized as follows:

- We propose a novel Spatiotemporal Distilled Dense-Connectivity Network (SDDN) for action recognition.
- We propose to generalize dense-connectivity into spatiotemporal domain via building block-level dense connections between appearance and motion streams, permitting effective spatiotemporal interaction at the feature representation layers.
- We propose a novel knowledge distillation module, which is composed of two students and a teacher, allowing appearance and motion streams to interact effectively at the high level layers.
- Our model obtains promising performance in action recognition on two benchmark datasets, including UCF101 [22] and HMDB51 [23] respectively.

The rest of this paper is organized as follows. Section 2 briefly reviews some related works. In Section 3, we describe our Spatiotemporal Distilled Dense-Connectivity Network (STDDCN) in detail. Experimental results are presented in Section 4 and some discussions are illustrated in Section 5. Finally, in Section 6, we conclude the paper.

2. Related works

In this section, we will review some works closely related to our STDDCN, including action recognition and knowledge distillation.

2.1. Action recognition

Video-based action recognition has been extensively studied and can be roughly divided into three categories.

The first category of action recognition approaches attempted to extract the spatiotemporal features from optical flow-based motion information by crafting, including Motion Boundary Histograms (MBH) [24], trajectories [2] and Histogram of Flow (HOF) [25], or via spatiotemporal oriented filtering, such as Cuboids [26], SOEs [27,28] and HOG3D [29].

The second group of action recognition methods concentrated on learning spatiotemporal features end-to-end, leveraging the breakthroughs [30] in image classification with Convolutional Neural Networks(CNNs) [31]. Among them, some work focused on the use of unsupervised learning [32,33]. Other work explored to combine the learned and hand-crafted features together [34]. Conversely, an alternative 3D spatiotemporal ConvNet was proposed to directly learn both spatial and temporal filter kernels [8]. Another research line focused on aggregating temporal information over extended time, such as temporal pooling of convolutional layers [35], weighted averaging of video frames over time or longer convolutions across time [36]. Moreover, to further model the temporal structure effectively, some researchers have incorporated LSTMs into their action recognition frameworks [4,37–40].

Taking inspiration from neuroscience, the third category of action recognition methods introduced two-stream ConvNet architecture [7,16–18], to extract RGB and Flow information in parallel. The final action classification score was obtained by fusing the scores of two streams. In [16], Wang et al. proposed Temporal Segment Network (TSN) for video-based action recognition, aiming at modeling long-range temporal structure underlying actions. To further improve the action recognition performance, many extensions of two-stream ConvNet [7] which investigate residual connections [17] and convolutional fusion [18] were proposed. Similar to our work, the model in [17] also builds connections among appearance and motion streams. Differently, our STDDCN leverages multiscale information by dense-connectivity interaction. Moreover, STDDCN contains a novel knowledge distillation module which allows appearance and motion streams to interact more effectively.

To better model the long-range temporal structure underlying actions, our model is built on the top of the promising TSN architecture [16].

2.2. Knowledge distillation

Recent work has also explored how to adopt additional information (or 'knowledge') to facilitate the training of the specific deep neural networks (DNN). In [41], Bucila et al. first proposed to utilize a single neural network to approximate an ensemble of classifiers. Recently, Hinton et al. developed a framework

to distill knowledge [21], in this scenario the predicted distribution, is distilled from a large teacher network into a smaller student network. Also, Hu et al. develop a teacher-student architecture to distill massive knowledge sources, containing logic rules, into DNNs [42,43]. To explore more diverse knowledge in intermediate feature maps, FitNets [44] and Attention Transfer [45] have been developed. Moreover, a unique type of knowledge inside deep metric learning model was also proposed [46] to train the student network.

In summary, knowledge distillation is an effective approach to distill complementary information from a teacher network to the student network, giving us clear grounds to introduce it into our model. For action recognition, appearance and motion streams are known to contain complementary information. Moreover, the fusion of two streams in soft probability output layer can be treated as a novel distribution of video data, which also carries complementary information to both two streams. Thus, we propose a novel knowledge distillation module and attach it to our framework, to realize the effective interactions among two streams by thoroughly leveraging the complementary information among them.

2.3. Feature representation

From the aspect of acquiring feature representation from multiple sources, including multimodal, multi-layer, multivariate and multi-task, some works are related to our STDDCN. Specifically, in [47], Hong et al. proposed a new 3D human pose recovery method, with feature extraction based on multimodal fusion (including representations from silhouettes and Mocap data). In [48], Wang et al. developed a novel type Multiple Instance Neural Networks (MINNs) to learn bag representations for multiple instance learning, including MI-Net, MI-Net with deep supervision (MI-Net-DS) and MI-Net with residual connections (MI-Net-RC) models respectively. Among them, the feature representations of MI-Net-DS and MI-Net-RC are based on multi-layered information and obtain better performance than that of MI-Net. Du et al. [49] proposed a new hierarchical deep neural network (HDNN) to handle the multivariate regression problem, which was realized via transferring the original problem to multiple subproblems. In [50], Yu et al.developed a promising image privacy protection method, which is based on the joint learning of deep CNN and tree classifier via multi-task learning strategy. Similar with above mentioned methods, feature representation of our method is also from multiple sources. However, our feature extraction relies on different modalities of the same data, including RGB and Flow modalities.

3. Spatiotemporal distilled dense-Connectivity network

In this section, we will depict the proposed STDDCN in detail (the pipeline is presented in Fig. 1). STDDCN is mainly composed of two densely connected block-level subnetworks and one knowledge distillation module. The purpose of STDDCN is to explore effective hierarchical spatiotemporal interactions among appearance and motion streams derived from the source video.

3.1. Baseline architecture

The baseline architecture of STDDCN is built on the top of Temporal Segment Network (TSN) [16], which aims at modeling long-term temporal structure. TSN contains two stream ConvNets, including appearance and motion stream ConvNets respectively. It works on a sequence of short clips sparsely collected from the whole video, other than depending on frame stacks or single frames. Each clip will obtain its own action recognition prediction and the fusion of these predictions form the final video-level prediction. In detail, each short clip contains an RGB image for appearance stream and a stack of $L = 10$ vertical and horizontal optical flow frames for motion stream.

Based on TSN, our model STDDCN establishes block-level dense connections from motion stream convNet to the appearance one, for obtaining spatio-temporal interaction at the feature extraction layers. In addition, STDDCN also integrates a novel knowledge distillation module for achieving high-level spatio-temporal interaction. Specifically, the ConvNet adopted here is DenseNet [20]. Fig. 2 illustrates the architectures of TSN and STDDCN.

3.2. Dense-Connectivity across two ConvNets

In a conventional TSN [16] based action recognition frameworks, the appearance and motion streams have no interactions except the last fusion of their softmax prediction layer [16]. Thus, truly spatiotemporal features cannot be extracted in their design since their is a lack of earlier interactions among two streams during processing.

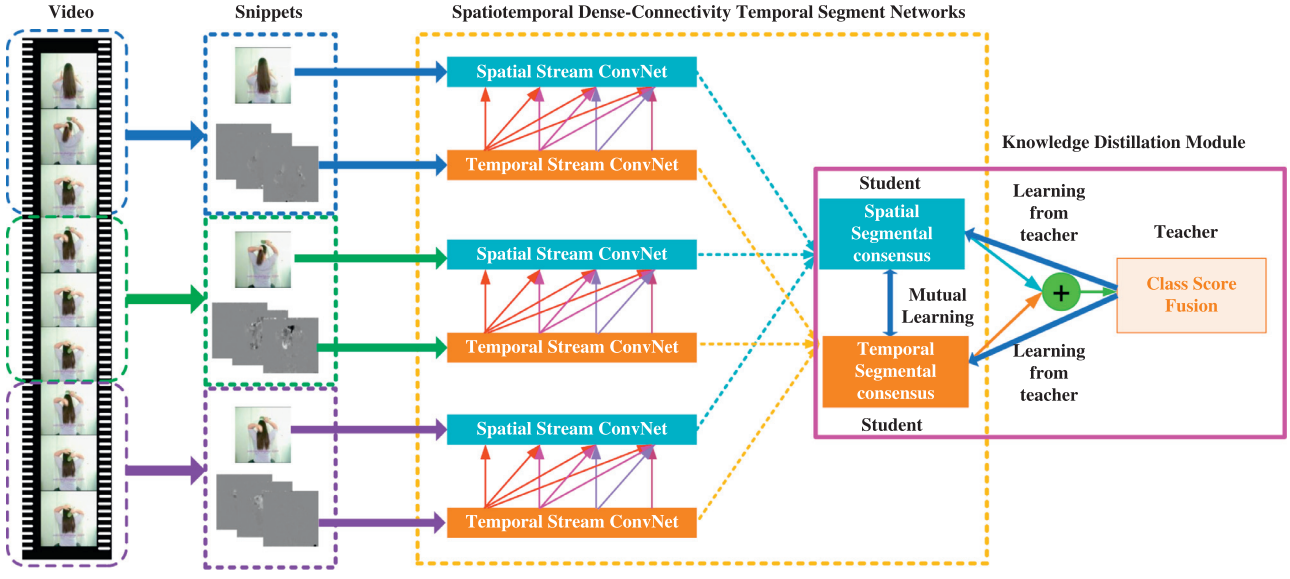

Fig. 1. The basic pipeline of our Spatiotemporal Distilled Dense-Connectivity Network (STDDCN) for video action recognition.

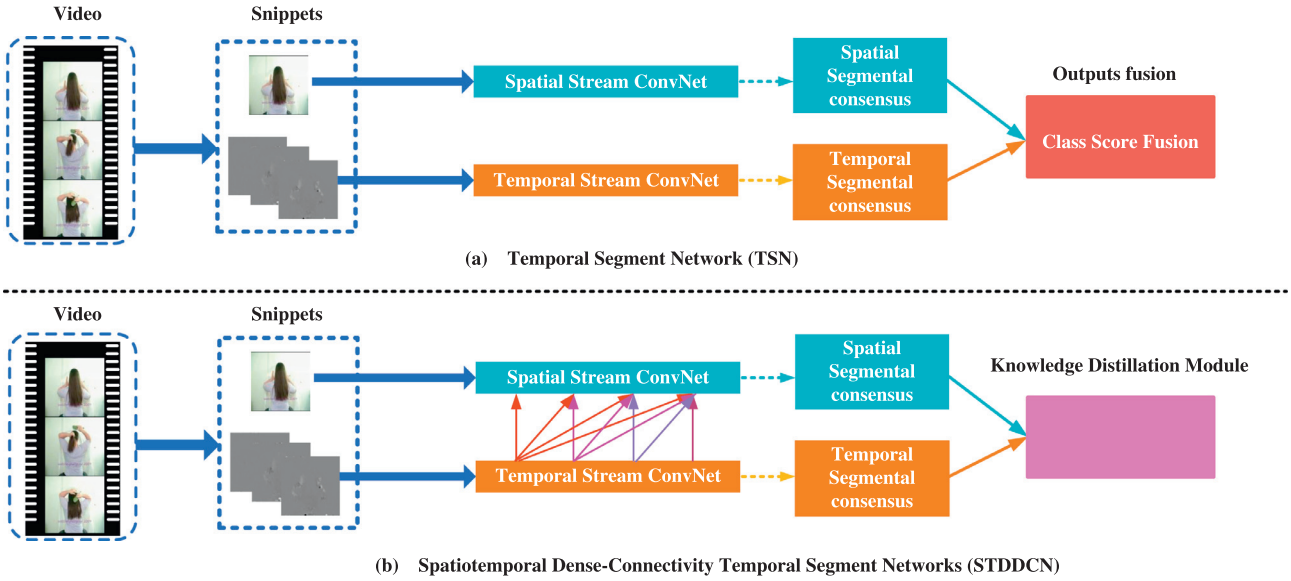

Fig. 2. Architectures of TSN and STDDCN. For simplicity, only one video snippet is presented.

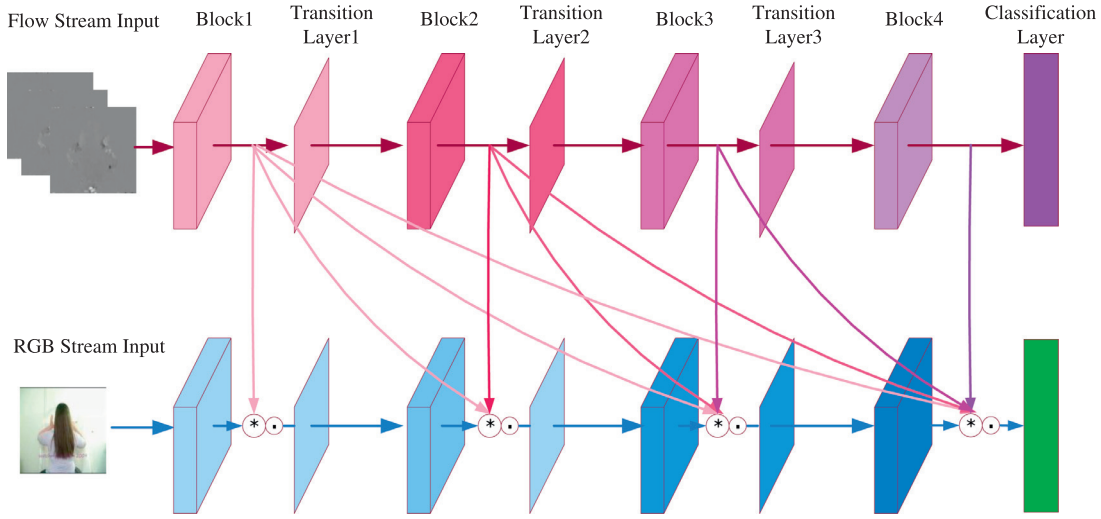

Fig. 3. Block-level dense-connectivity across two ConvNets. The inputs of the current block of appearance stream is the feature maps of all preceding blocks from the motion stream and the feature map of the former block of the appearance stream.

Dense Convolutional Network (DenseNet) [20], a recently proposed deep learning architecture, utilizes feature maps of all preceding layer as the current layer's input. It surpasses other frameworks in terms of relieving vanishing-gradient problem, enhancing feature propagation and encouraging feature reuse.

Consequently, we generalize the appealing dense-connectivity of DenseNet into the spatiotemporal domain, hoping to build effective earlier interactions among appearance and motion streams. The detailed structure of this module is presented in Fig. 3, with block-level dense connections established among two-stream DenseNets. Insights of this design are that, on the one hand, block-level dense-connectivity can encourage spatiotemporal interactions among two streams. On the other hand, it can guarantee the information specificity of two streams in some extent. As detailed above, connections built here are unidirectional from motion stream to appearance stream.

Concretely, block-level dense-connectivity built here can be formulated as:

$$X_R^{i+1} = f\left(X_R^i\right) + G\left(f(X_R^i), X_F^i\right) \tag{1}$$

where X_R^i and X_F^i denote the inputs for *ith* block of appearance and motion ConvNets correspondingly. f indicates the original function that transfers the input of *ith* block to $i+1th$ block in the corresponding ConvNet, $+$ denotes the elementwise addition. In Eq. (1), elementwise addition is utilized to do information fusion, which aims at injecting the information of preceding motion ConvNet layers X_F^i into the current appearance ConvNet layer X_R^{i+1}. G denotes the multiplicative gate, which is employed to modulate the appearance features by the motion signal, and can be depicted as:

$$G\left(f\left(X_R^i\right), X_F^i\right)\right) = f\left(X_R^i\right) * \left[H^i(X_F)\right)\right] \tag{2}$$

where $*$ denotes elementwise multiplication and the modulation is realized by it. The elementwise multiplication will force two streams to interact in both feedforward and feedback passes (see Eqs. (3)–(5)). In addition, $[H^i(X_F))]$ indicates the concatenation of all preceding blocks' feature maps of motion stream, which can be represented as

$$\left[H^i(X_F)\right)\right] = \left[H_1^i\left(X_F^1\right), \cdots, H_i^i\left(X_F^i\right)\right] \tag{3}$$

where H_j^i represents a weight matrix that transfer X_F^j to the same size with X_R^i.

Eq. (1) illustrates that the input of the $i+1$ block of appearance ConvNet is the integration of two streams. One is the input $f(X_R^i)$ from the original appearance ConvNet and the other is the multiplicative gate fusion of $f(X_R^i)$ and feature maps of all preceding blocks $[(X_F^1), \cdots, (X_F^i)]$ from motion ConvNet.

Based on the above formulation, gradient of the loss function $\mathcal{L}$ during its backward processing can be demonstrated as:

$$\frac{\partial \mathcal{L}}{\partial X_R^i} = \frac{\partial \mathcal{L}}{\partial X_R^{i+1}} \frac{\partial X_R^{i+1}}{\partial X_R^i} = \frac{\partial \mathcal{L}}{\partial X_R^{i+1}} \left(\frac{\partial f\left(X_R^i\right)}{\partial X_R^i} + [H^i(X_F)] \frac{\partial G(.)}{\partial X_R^i} \right) \tag{4}$$

$$\frac{\partial \mathcal{L}}{\partial X_F^i} = \frac{\partial \mathcal{L}}{\partial X_F^{i+1}} \frac{\partial X_F^{i+1}}{\partial X_F^i} + f(X_R^i) \frac{\partial \mathcal{L}}{\partial X_R^{i+1}} \frac{\partial G(.)}{\partial X_F^i} \tag{5}$$

where $G(.)$ indicates $G(f(X_R^i), [H^i(X_F)]))$.

Based Eqs. (4) and (5), gradients passing through appearance and motion streams are not only adjusted by the information from their own stream, but also modulated by the other stream. Concretely, gradient of appearance stream is partially adjusted by the motion information $[H^i(X_F)]$, and the motion stream's gradient is partially modulated by the appearance information $f(X_R^i)$.

3.3. Knowledge distillation module

To realize the interactions across appearance and motion streams in the video at the high level layers, we propose a novel knowledge distillation module. It contains two students and a teacher. Specifically, the output probabilities of two streams are treated as two students and the fusion of two streams' output is seen as the teacher. Specifically, one student is modulated by the other student and the teacher. For simplicity, two students are dubbed as *stu*1, *stu*2 and the teacher is represented as *tea*. Under this setting, the knowledge distillation loss $\mathcal{L}_{s1}$ for *stu*1 can be represented as:

$$\mathcal{L}_{s1} = \beta * \mathcal{L}_{dis_{s1}^t} + (1-\beta) * \mathcal{L}_{dis_{s1}^{s2}} \tag{6}$$

where $\mathcal{L}_{dis_{s1}^t}$ and $\mathcal{L}_{dis_{s1}^{s2}}$ indicate the distillation losses from the teacher *tea* and the other student2 (*stu*2) to student1 (*stu*1) respectively. β is the hyperparameter controlling the strengths of two loss terms. The definition of $\mathcal{L}_{dis_{s1}^t}$ and $\mathcal{L}_{dis_{s1}^{s2}}$ are denoted as:

$$\begin{aligned} \mathcal{L}_{dis_{s1}^t} &= \alpha * \mathcal{L}_{kl}^{(s1,t)} + (1-\alpha) * \mathcal{L}_{cro}^{s1} \\ \mathcal{L}_{dis_{s1}^{s2}} &= \alpha * \mathcal{L}_{kl}^{(s1,s2)} + (1-\alpha) * \mathcal{L}_{cro}^{s1} \end{aligned} \tag{7}$$

where $\mathcal{L}_{kl}^{(s1,t)}$ indicates the Kullback-Leibler divergence about *stu*1 and *tea* and $\mathcal{L}_{kl}^{(s1,s2)}$ has the similar meaning. $\mathcal{L}_{cro}^{s1}$ denotes the cross entropy loss for *stu*1. α indicates the hyperparameter balancing the strengths of two corresponding loss terms. Among them, $\mathcal{L}_{kl}^{(s1,t)}$ and $\mathcal{L}_{kl}^{(s1,s2)}$ can be formulated as:

$$\begin{aligned} \mathcal{L}_{kl}^{(s1,t)} &= D_{kl}\left(X^{s1}, X^t\right) = D_{kl}\left[P\left(\bar{Y}^t/T\right)||P\left(\bar{Y^{s1}}/T\right)\right. \\ &= \sum_{i\in\mathcal{S}} P\left(\bar{Y}_i^t/T\right) log \frac{P\left(\bar{Y}_i^t/T\right)}{P\left(\bar{Y}_i^{s1}/T\right)} \\ \mathcal{L}_{kl}^{(s1,s2)} &= D_{kl}\left(X^{s1}, X^{s2}\right) = D_{kl}\left[P\left(\bar{Y^{s2}}/T\right)||P\left(\bar{Y^{s1}}/T\right)\right. \\ &= \sum_{i\in\mathcal{S}} P\left(\bar{Y}_i^{s2}/T\right) log \frac{P\left(\bar{Y}_i^{s2}/T\right)}{P\left(\bar{Y}_i^{s1}/T\right)} \end{aligned} \tag{8}$$

where X^{s1}, X^{s2} and X^t denote the input samples of *stu*1, *stu*2 and *tea* networks respectively. $\bar{Y^{s1}}$, $\bar{Y^{s2}}$ and $\bar{Y^t}$ represent the output probabilities of the *stu*1, *stu*2 and *tea* networks correspondingly. T indicates the temperature parameter. S denotes the set of all training samples and i represents the index of the *ith* sample. $P(Z_i)$ can be represented as:

$$P(Z_i) \quad = exp(Z_i)/\textstyle\sum_{j\in\mathcal{S}} exp(Z_j) \tag{9}$$

The cross entropy loss $\mathcal{L}_{cro}$ can be formulated as:

$$\begin{aligned} \mathcal{L}_{cro}^{s1} &= D_{cro}\left(X^{s1}, Y^{s1}\right) \\ &= -\sum_{i\in\mathcal{S}} \left(Y_i^{s1} * log \frac{exp(\bar{Y_i^{s1}})}{1+exp(\bar{Y_i^{s1}})} + (1-Y_i^{s1}) * log \frac{1}{1+exp(\bar{Y_i^{s1}})} \right) \end{aligned} \tag{10}$$

where X^{s1}, $\bar{Y^{s1}}$, $\mathcal{S}$ and i share the same meanings with those defined in Eq. (8). Y_i^{s1} indicates the target label for input sample X_i^{s1}.

Similar with $\mathcal{L}_{s1}$, the computation of $\mathcal{L}_{s2}$ can be given as follows:

$$\mathcal{L}_{s2} = \beta * \mathcal{L}_{dis_{s2}^t} + (1-\beta) * \mathcal{L}_{dis_{s2}^{s1}} \tag{11}$$

where $\mathcal{L}_{dis_{s2}^t}$ and $\mathcal{L}_{dis_{s2}^{s1}}$ indicate the distillation losses from the teacher *tea* and the other student *stu*1 respectively. β is the hyperparameter controlling the strengths of two loss terms. The definition of $\mathcal{L}_{dis_{s2}^t}$ and $\mathcal{L}_{dis_{s2}^{s1}}$ are described as:

$$\begin{aligned} \mathcal{L}_{dis_{s2}^t} &= \alpha * \mathcal{L}_{kl}^{(s2,t)} + (1-\alpha) * \mathcal{L}_{cro}^{s2} \\ \mathcal{L}_{dis_{s2}^{s1}} &= \alpha * \mathcal{L}_{kl}^{(s2,s1)} + (1-\alpha) * \mathcal{L}_{cro}^{s2} \end{aligned} \tag{12}$$

where $\mathcal{L}_{kl}^{(s2,t)}$ indicates the Kullback-Leibler divergence about *stu*2 and *tea* and $\mathcal{L}_{kl}^{(s2,s1)}$ has the similar meaning. $\mathcal{L}_{cro}^{s2}$ denotes the cross entropy loss for *stu*2. α indicates the hyperparameter balancing the strengths of two corresponding loss terms. Among them, $\mathcal{L}_{kl}^{(s2,t)}$ and $\mathcal{L}_{kl}^{(s2,s1)}$ can be formulated as:

$$\begin{aligned} \mathcal{L}_{kl}^{(s2,t)} &= D_{kl}\left(X^{s2}, X^t\right) = D_{kl}\left[P\left(\bar{Y}^t/T\right)||P\left(\bar{Y^{s2}}/T\right)\right. \\ &= \sum_{i\in\mathcal{S}} P\left(\bar{Y}_i^t/T\right) log \frac{P\left(\bar{Y}_i^t/T\right)}{P\left(\bar{Y}_i^{s2}/T\right)} \\ \mathcal{L}_{kl}^{(s2,s1)} &= D_{kl}\left(X^{s2}, X^{s1}\right) = D_{kl}\left[P\left(\bar{Y^{s1}}/T\right)||P\left(\bar{Y^{s2}}/T\right)\right. \\ &= \sum_{i\in\mathcal{S}} P\left(\bar{Y_i^{s1}}/T\right) log \frac{P\left(\bar{Y_i^{s1}}/T\right)}{P\left(\bar{Y_i^{s2}}/T\right)} \end{aligned} \tag{13}$$

where X^{s1}, X^{s2}, X^t, $\bar{Y^{s1}}$, $\bar{Y^{s2}}$, $\bar{Y^t}$, T, S and i share the same meanings with those defined in Eq. (8). The cross entropy loss $\mathcal{L}_{cro}^{s2}$ can be formulated as:

$$\begin{aligned} \mathcal{L}_{cro}^{s2} &= D_{cro}\left(X^{s2}, Y^{s2}\right) \\ &= -\sum_{i\in\mathcal{S}} \left(Y_i^{s2} * log \frac{exp\left(\bar{Y_i^{s2}}\right)}{1+exp\left(\bar{Y_i^{s2}}\right)} + \left(1-Y_i^{s2}\right) * log \frac{1}{1+exp\left(\bar{Y_i^{s2}}\right)} \right) \end{aligned} \tag{14}$$

where X^{s2}, $\bar{Y^{s2}}$, $\mathcal{S}$ and i share the same meanings with those defined in Eq. (8). Y_i^{s2} indicates the target label for input sample X_i^{s2}.

The final loss utilized for training the whole network is represented as:

$$\mathcal{L} = \mathcal{L}_{s1} + \mathcal{L}_{s2} \tag{15}$$

Table 1
Comparison results of different models on HMDB51 and UCF101.

Dataset	HMDB51						UCF101	
Split	Split1		Split2		Split3		Split1	
Model	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN
RGB	52.55	58.56	49.02	56.01	49.61	57.19	84.00	86.23
Flow	57.45	56.80	57.65	56.21	61.37	61.18	85.93	86.36
Two	66.73	**67.52**	64.77	**66.07**	65.56	**66.95**	93.46	**93.78**

4. Experiments

4.1. Datasets and implementation details

Two popular benchmark datasets, including HMDB51 [23] and UCF101 [22], are utilized to verify the superiority of our proposed STDDCN. UCF101 contains 101 action classes and 13, 320 video clips, whose evaluation scheme follows that of THUMOS13 challenge [51]. In addition, HMDB51 is collected from the realistic videos that include various sources, such as web videos and movies, which contains 6, 766 video clips and 51 action classes. All our experiments follow the original evaluation scheme that utilizing three training/testing splits, to validate the performance of corresponding models.

We use stochastic gradient descent algorithm (SGD) to train our models, with a total mini-batch size 32. All models are first initialized by the pre-trained models based on ImageNet [52]. The learning rate starts with 0.0001 and then decreased to its $\frac{1}{10}$ per 12,000 iterations. The momentum is 0.9. The α and β are 0.1 and 0.9 respectively. We stop our training at 20, 000 iterations. In order to avoid over-fitting, the following data augmentation strategies are adopted, including location jittering, corner cropping, horizontal flipping and scale jittering. Our experiments are executed on TITANX GPUs.

4.2. Experimental results

4.2.1. Evaluation of proposed STDDCN

To examine the performance of our developed model STDDCN, we compare it with the baseline model TSN [16] on HMDB51 [23] and UCF101 [22] and other state-of-the-art models, with comparison results displayed in Table 1. Specifically, in Table 1, RGB, Flow and Two indicate the action recognition accuracies of different models based on appearance ConvNet, motion ConvNet and the fusion of two ConvNets respectively. The ConvNet utilized here is DenseNet121. In the following sections, without statement, basic ConvNet is defaulted as DenseNet121.

From Table 1, we can see that the STDDCN yields consistent better results than other model, which verifies the superiority of STDDCN in all splits of HMDB51 and the first split of UCF101. On the other hand, our model performs worse than TSN when only flow network is adopted. Reasons can be summarized as follows. STDDCN jointly trains two branch networks, and the parameters are iteratively updated to force the model to achieve the optimal fusion results. The one branch contributed more to the final fusion will obtain more emphasis and achieve better performance. From Table 1, we can see that for HMDB51, the RGB network obtains better results than those of TSN, otherwise the flow network is worse. In addition, RGB network also obtains superior performance than its Flow counterpart. Concerning UCF101, both RGB and Flow networks obtains better results than those of TSN and two networks achieve comparable performance. These results indicate that the RGB stream of HMDB51 contains more discriminant information, and both RGB and Flow streams of carry important cues.

4.2.2. Evaluation of Alpha (α) and Beta (β) in Knowledge Distillation (KD) module

To validate the effects of α and β in KD module, we perform action recognition with various α and β ranges from 0.1 to 0.9 with step size 0.2 in KD module. Results are exhibited in (a) and (b) of Fig. 4. As alpha increases, the action recognition performance degrades. In addition, we also report the result of model that does not perform knowledge distillation which $\alpha = 0$. From (a) of Fig. 4, we can see that the model with $\alpha = 0.1$ obtains the best result in all splits of HMDB51, again validating the effectiveness of knowledge distillation module. In addition, as shown in (b) of Fig. 4, although beta β has no obvious trend of changing in a certain direction, it obtains promising result with $\beta = 0.3$ in all splits. Thus, α and β are defaulted as 0.1 and 0.3 respectively in the following experiments.

4.2.3. Evaluation of Temperature (T) in KD module

Besides α and β, temperature T is also an important parameter in KD module. To examine the effects of T, we compare models with different T ranges from 2 to 5 on HMDB51 and UCF101. Results are shown in (c) of Fig. 4. The impact of temperature T to action recognition is not obvious with its value ranges from 2 to 5. Moreover, T with value 4 offers the best performance in most cases. Thus, T is defaulted as 4.

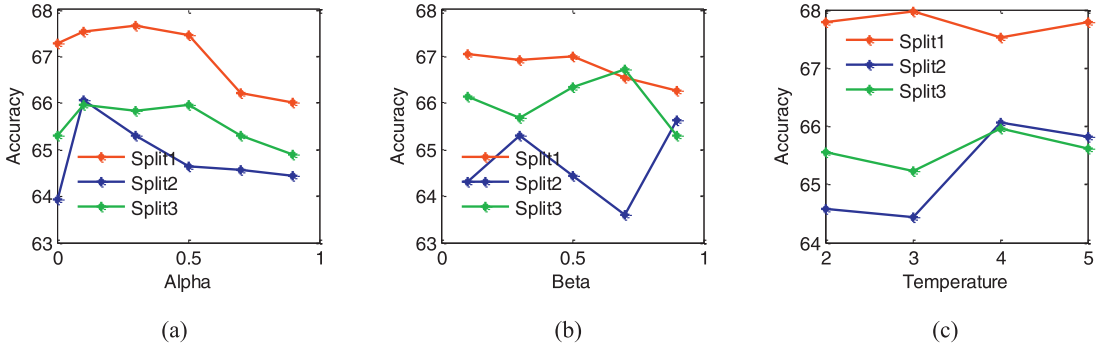

Fig. 4. Comparison results based on models with different alpha, beta and temperature values in the KD module on HMDB51.

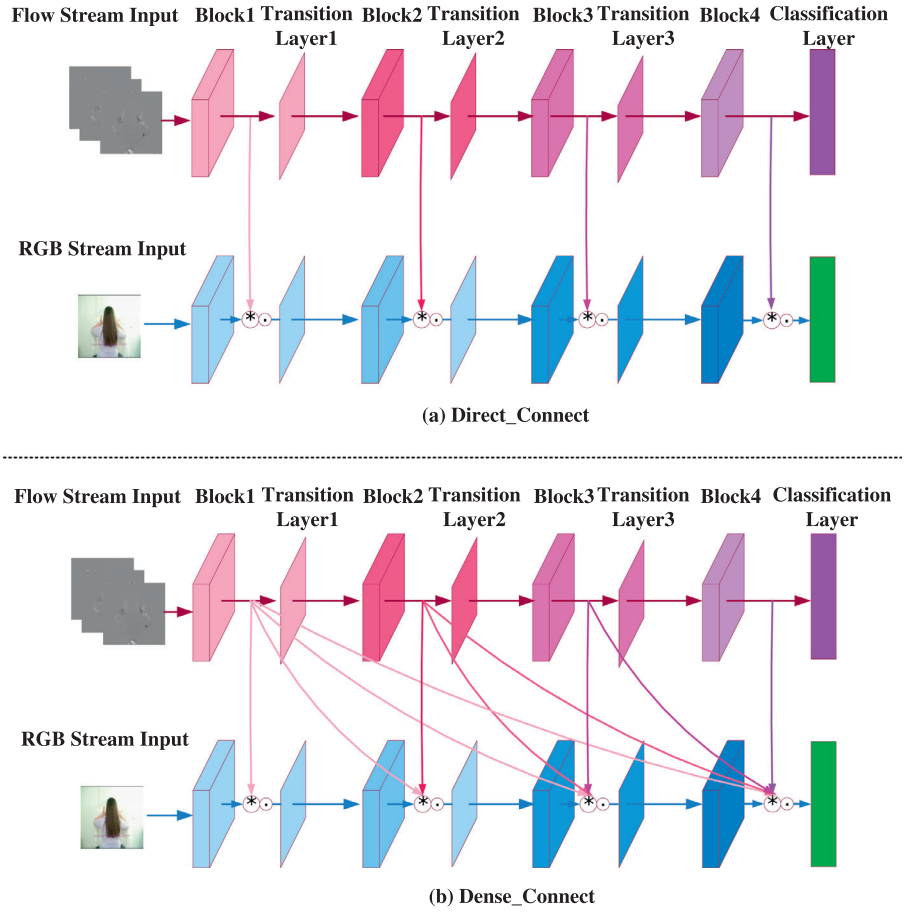

Fig. 5. Different connection approaches across two ConvNets.

Table 2
Comparison of different connection models on HMDB51 and UCF101.

Dataset	HMDB51						UCF101	
Split	Split1		Split2		Split3		Split1	
Model	Dir_Con	Den_Con	Dir_Con	Den_Con	Dir_Con	Den_Con	Dir_Con	Den_Con
RGB	52.94	58.76	50.98	55.03	53.59	56.67	84.87	86.33
Flow	57.32	57.65	55.56	56.21	60.13	60.52	85.97	86.50
Two	66.08	**67.04**	63.39	**64.64**	65.16	**65.29**	92.59	**93.22**

4.2.4. Evaluation of dense connection

To explore the effects of dense connections between appearance and motion streams, comparisons are made among the following models, including Direct_Connect and Dense_Connect (Presented in Fig. 5). Note that, these models have no knowledge distillation module. Concerning Direct_Connect model, it integrates feature maps of the current block from motion stream into the corresponding block of the appearance stream, as shown in (a) of Fig. 5. While for Dense_Connect model, it fuses feature maps of all preceding blocks, as shown in (b) of Fig. 5. Comparison results are presented in Table 2, with Direct_Connect and Dense_Connect models dubbed as Dir_Con and Den_Con respectively for simplicity.

Table 2 shows that the results of Dense_Connect model are uniformly better than those of Direct_Connect model, which validates the effectiveness of dense connection.

4.2.5. Evaluation of knowledge distillation (KD)

To examine the effectiveness of proposed knowledge distillation module, we make comparisons among the following three variants, containing Distill_s, Distill_t and Distill_st (Illustrated in Fig. 6). Specifically, Distill_s indicates the model that two student ConvNets performing mutual learning. In another word, each student is taught by the knowledge learned from the other student. Distill_t refers the model that two student ConvNets are only taught by the teacher ConvNet correspondingly. In addition, Distill_st denotes the model that each student is not only modulated by the other student but also by the teacher. Detailed comparison results are exhibited in Table 3. In Table 3, Distill_s, Distill_t and Distill_st dubbed as Dis_s, Dis_t and Dis_st correspondingly.

Table 3 demonstrates that Dis_st model achieves uniformly better results than those of Dis_t and Dis_s models, which indicates that the proposed knowledge distillation module with two students and a teacher is optimal.

4.2.6. Evaluation of the computational time

To further validate the efficiency of our model, we have compared the computational time (training and testing time) of different models, which is presented in Table 4. In Table 4, TSN-RGB indicates the individual RGB stream network in TSN framework, and TSN-Flow denotes the Flow stream counterpart, other models

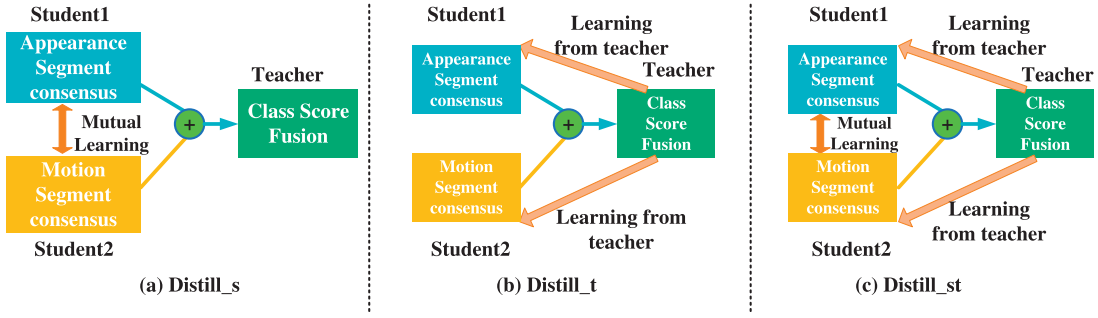

Fig. 6. Different knowledge distillation models.

Table 3
Comparison of different distillation models.

Dataset	HMDB51									UCF101		
Split	Split1			Split2			Split3			Split1		
Model	Dis_s	Dis_t	Dis_st	Dis_s	Dis_t	Dis_st	Dis_s	Dis_t	Dis_st	Dis_s	Dis_t	Dis_st
RGB	60.00	58.17	58.56	55.56	53.20	56.01	57.65	55.62	57.19	84.94	84.99	86.23
Flow	56.47	57.12	56.80	55.82	57.06	56.21	61.57	60.33	61.18	86.29	86.42	86.36
Two	66.67	67.45	**67.52**	65.09	64.70	**66.07**	65.49	65.29	**66.95**	93.74	93.41	**93.78**

Table 4
Comparison of computational time for different models.

Model	TSN-RGB	TSN-Flow	TSN	Dir_Con	Den_Con	Dis_s	Dis_t	Dis_st
Train_time(s)	0.386	0.395	0.781	0.803	0.808	0.807	0.809	0.810
Test_time(s)	0.187	0.193	0.380	0.378	0.381	0.379	0.380	0.380

Table 5
Comparison of models with different connection directions on HMDB51.

	HMDB51									UCF101		
Split	Split1			Split2			Split3			Split1		
Model	A ← M	A → M	A↔M	A ← M	A → M	A↔M	A ← M	A → M	A↔M	A ← M	A → M	A↔M
RGB	58.56	51.34	51.31	56.01	47.32	41.24	57.19	48.24	40.59	86.23	83.60	84.02
Flow	56.80	59.54	59.28	56.21	56.93	56.21	61.18	57.84	51.31	86.36	90.52	89.23
Two	**67.52**	60.23	59.89	**66.07**	58.10	57.45	**66.95**	58.36	51.99	**93.78**	90.65	**89.62**

Table 6
Comparison of various DenseNet on HMDB51 and UCF101.

Dataset	HMDB51						UCF101	
Split	Split1		Split2		Split3		Split1	
Den121	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN
RGB	52.55	58.56	49.02	56.01	49.61	57.19	84.00	86.23
Flow	57.45	56.80	57.65	56.21	61.37	61.18	85.93	86.36
Two	66.73	**67.52**	64.77	**66.07**	65.56	**66.95**	93.46	**93.78**
Den161	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN
RGB	54.44	58.89	51.93	56.92	52.22	56.41	86.84	87.47
Flow	57.58	57.84	57.19	58.43	59.93	61.11	88.08	87.65
Two	69.28	**70.20**	67.23	**68.95**	68.43	**69.34**	94.12	**94.79**

share the same meanings in the above section. From Table 4, we can see that the training time of all our model variants (Dir_Con, Den_Con, Dis_s, Dis_t, Dis_st) are a little more than that of the TSN. This is because although the whole parameters of our models are comparable with that of TSN, the connection allows our model to interact not only in its forward pass but also in its feedback pass, which leads to more training time. Moreover, the training time of Den_Con model is a slightly more than that of Dir_Con one, which is due to more connections existing in the Den_Con model. In addition, the training times of Dis_t, Dis_t, Dis_st models are almost the same with Den_Con model, as no more computations are introduced by them. Concerning the test time, our models (Dir_Con, Den_Con, Dis_s, Dis_t, Dis_st) are comparable with that of TSN, which is because there is no feedback interactions needed to perform in the test phase. We should note that the time in Table 4 refers to the computational time for one batch.

4.2.7. Evaluate the direction of dense connection

As stated above, dense connections between two streams are from flow stream to appearance stream. Reasons can be found in literature [53]. Concerning our specific dense-connectivity, effects of the directions of dense connection also be validated, with results presented in Table 5. Specifically, in Table 5, A ← M, A → M and A↔M indicate models with dense connection from motion to

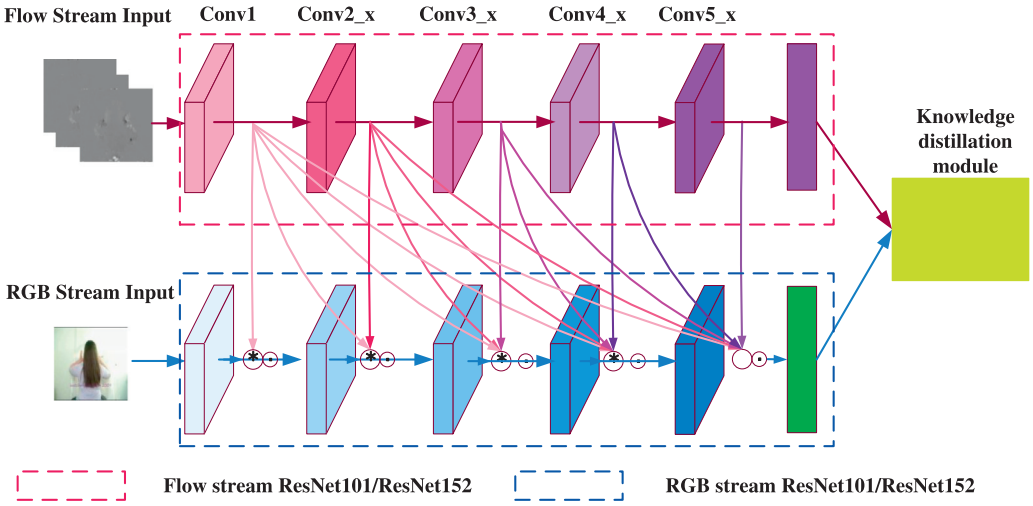

Fig. 7. STDDCN with ResNet basic network.

Table 7
Comparison of various ResNet models on HMDB51 and UCF101.

Dataset	HMDB51						UCF101	
Split	Split1		Split2		Split3		Split1	
Res101	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN
RGB	51.57	54.77	51.57	52.94	53.27	53.27	86.00	87.49
Flow	58.56	59.28	55.49	57.19	60.26	60.46	87.26	87.76
Two	68.17	**68.32**	64.59	**65.31**	66.14	**66.66**	93.91	**94.52**
Res152	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN	TSN	STDDCN
RGB	55.42	56.80	53.92	56.60	53.86	56.60	87.57	88.44
Flow	57.97	58.30	56.99	57.65	60.72	61.11	87.52	88.33
Two	69.81	**70.08**	67.29	**68.12**	67.86	**68.98**	93.92	**94.67**

appearance stream, appearance to motion stream and the combination of them respectively. Table 5 reflects models A ← M achieves the best results among three models, which verifies the effectiveness of connection direction from motion to appearance stream. This is consistent with previous findings in [53].

4.2.8. Evaluation of model with various depth

To assess whether STDDCN can generalize well to models with various depths or not, we report the results of STDDCN based on DensetNet, including DenseNet121 and DenseNet161 (dubbed as Den121 and Den161), in Table 6. Table 6 illustrates that STDDCN achieves uniformly better results than those of TSN in all cases, which verifies the generalization capacity of STDDCN in terms of model depth. In addition, STDDCN with deeper architecture offers better results in action recognition.

4.2.9. Evaluation of model with diverse architecture

To further validate the generalization capacity of STDDCN in terms of network architectures, we generalize STDDCN to ResNet101 and ResNet152 (dubbed as Res101 and Res152). Specifically, similar with DenseNet, ResNet101 and ResNet152 are also composed of several blocks. Thus, the STDDCN is generalized to ResNet via the following steps. First, the block-level dense connections are established between appearance ResNet and motion ResNet. In addition, the proposed knowledge distillation module is integrated into the final fusion layer. Fig. 7 illustrates the architectures of generalized STDDCN based on ResNet101 and ResNet152. Comparison results are displayed in Table 7.

From Table 7, we can see that our model consistently surpasses the baseline model TSN, which validates the generalization capacity of STDDCN on various architectures. Moreover, results of basic architecture with ResNet152 are superior to those of ResNet101, again verifying the superiority of deeper model when performing specific tasks.

Table 8
Comparison with current state-of-the-art methods on UCF-101 and HMDB51 dataset.

Model	UCF101	HMDB51
IDT [2]	85.9	57.2
IDT(higher-dimension) [54]	87.9	61.1
MIFS(L=3) [55]	89.1	65.1
TDD [56]	-	63.2
KVMF [57]	-	63.3
VGG16+Images on Web [58]	83.5	-
Two-stream(fusion by averaging) [7]	86.9	-
Two-stream(fusion by SVM) [7]	88.0	59.4
Fusion Two-stream [18]	91.8	64.6
Action-transformations [59]	-	63.4
Two-stream(VGG-16) [16]	91.4	-
LRCN(weighted average) [60]	82.9	-
C3D(1 net+SVM) [8]	82.3	-
C3D(3 net+SVM) [8]	85.2	-
C3D+IDT [8]	90.4	-
T-CNN [61]	87.5	-
FstCN(averaging fusion) [62]	87.9	58.6
FstCN(SCI fusion) [62]	-	59.1
Asymmetric 3D-CNN(RGBF) [12]	87.7	61.2
Asymmetric 3D-CNN(RGB+RGBF) [12]	89.5	63.5
Asymmetric 3D-CNN(RGB+RGBF+IDT) [12]	92.6	65.4
TSN [16]	93.46	65.69
TSI3D [63]	93.4	66.4
Our	**93.78**	**66.87**

Table 9
Comparison of different spatiotemporal connection models.

Model	STR [17]	STM [53]	Our(Den121)	Our(Den161)	Our(Res101)	Our(Res152)
HMDB51	66.4	68.9	66.51	**69.49**	66.43	69.03
UCF101	93.4	94.2	93.78	**94.79**	94.52	94.67

4.2.10. Compare with other spatiotemporal architecture

Similar with our model, previous works [17,53] build residual connections among appearance and motion streams, called Spatiotemporal_ResNet and Spatiotemporal_Multiplier respectively. Our models differs from them in two folds. On the one hand, dense-connectivity in our STDDCN can leverage multi-scale information. On the other hand, STDDCN attaches a novel knowledge distillation module which can build effective spatiotemporal at the high level layers. Comparison results are shown in Table 9 (Spatiotemporal_ResNet and Spatiotemporal_Multiplier are dubbed as STR and STM in the table). Table 9 illustrates that our model obtains better results than Spatiotemporal_ResNet, verifying the superiority of our model.

4.2.11. Compare with the state-of-the-Art methods

In this section, we compare the STDDCN with recent proposed state-of-the-art approaches, with results presented in Table 8. Table 8 shows that when compared with the models based on trajectory features (IDT), two streams, C3D, TSN and some other methods, our model achieves the best performance, which further validates the spatiotemporal feature obtained by efficient spatiotemporal interactions is essential to the action recognition.

5. Discussions

5.1. Dense-Connectivity via multiplicative gate is vital

Most conventional two-stream based action recognition approaches train the appearance and motion streams independently. Few of them consider the interactions between two streams except the last fusion layer. Thus, truly spatiotemporal features cannot be extracted in their design. Targeting to tackle this problem, block-level dense-connectivity across appearance and motion streams are built to encourage earlier spatiotemporal interactions. Specifically, feature maps of all preceding blocks from the flow stream are integrated into the current block of appearance stream, via multiplicative gate. This unidirectional fusion design is attribute to that spatial stream dominates motion stream during training. Experimental results validate that models with dense-connectivity is superior. Reasons can be summarized as several folds. Firstly, multiplicative gate encourages network fusion from the first-order expanded to the second-order. Secondly, during both forward pass and gradient backward, appearance and motion representations are all modulated by signals from two paths, allowing two streams to interact effectively. Moreover, feature representation can be enhanced since all preceding blocks with multi-scale information have been leveraged.

5.2. Knowledge distillation with students and teacher is optimal

The developed knowledge distillation module is comprised of two students and a teacher. Concretely, two students refer to the output probabilities of appearance and motion streams, and the teacher represents the fusion output of two streams. To verify the effectiveness of the proposed knowledge distillation module, three variants are developed, including Distill_s (modulate one student only by knowledge learned from the other student), Distill_t(modulate one student only by knowledge learned from the teacher) and Distill_st(modulate one student by knowledge learned from both teacher and the other student simultaneously). Experimental results reveals that Distill_st yields the best results when compared to the Distill_s and Distill_t. Reasons can be summarized as follows: appearance and motion streams contain mutual complementary information to each other. On the other hand, the fusion of two streams, can be seen as a new distribution of data, which also carries complementary information to both appearance and motion streams. Specifically, concerning a student in Distill_st, the knowledge transferred to it not only from the other stream, but also from the new distribution of data derived by the fusion of two streams. Conversely, Distill_s and Distill_t only leverage one kind of complementary information either from the other student or from the new distribution of data. Thus, knowledge distillation module with two students and a teacher leverages the most complementary information underlying appearance and motion streams and is optimal.

5.3. STDDCN Can generalize well

To verify the generalization capacity of our model STDDCN in terms of different network depths and structures, STDDCN is executed based on the following basic models. They are DenseNet (with two variants of different depths: DenseNet121 and DenseNet161) and ResNet (with two variants of different depths: ResNet101 ResNet152) respectively. Experimental results show the performances of STDDCN are uniformly better than those of the baseline model TSN, under different network depths and structures settings, which validates the excellent generalization capability of STDDCN. Moreover, experimental results also verify the superiority of deeper networks in performing action recognition.

6. Conclusions

This paper proposes a novel Spatiotemporal Distilled Dense-Connectivity Network (STDDCN) for action recognition, which is comprised of two densely connected subnetworks and a knowledge distillation module. The block-level dense-connectivity among appearance and motion streams encourages effective spatiotemporal interaction at the feature representation layer. Moreover, knowledge distillation module, which consists of two students and a teacher, facilitates the spatiotemporal interaction at the high-level layers by thoroughly leveraging the complementary information underlying two streams. In summary, STDDCN allows effective hierarchical spatiotemporal interactions between appearance and motion streams. Moreover, it can be trained end-to-end. Ablative studies based on the benchmark datasets UCF101 and HMDB51 verify the effectiveness and generalization of STDDCN. Experimental results validate that STDDCN obtains superior action recognition performance, when compared to the conventional two-stream based action recognition approaches. Future works will explore more effective interaction strategies across appearance and motion streams for improved action recognition.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (No. 2018YFB1004600), the Beijing Municipal Natural Science Foundation (No. Z181100008918010), the National Natural Science Foundation of China (No. 61761146004, No. 61773375,

No. 61836014), and in part by the Microsoft Collaborative Research Project.

References

[1] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2–3) (2005) 107–123.
[2] H. Wang, C. Schmid, Action recognition with improved trajectories, in: International Conference on Computer Vision, 2013, pp. 3551–3558.
[3] L. Wang, Y. Qiao, X. Tang, Motionlets: Mid-level 3d parts for human motion recognition, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2013, pp. 2674–2681.
[4] L. Wang, Y. Qiao, X. Tang, Mofap: a multi-level representation for action recognition, Int. J. Comput. Vis. 119 (3) (2016) 254–271.
[5] T.D. Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, D. Windridge, An evaluation of bags-of-words and spatio-temporal shapes for action recognition, in: Proceedings of the Winter Conference on Applications of Computer Vision, 2011, pp. 344–351.
[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
[7] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, Advances in Neural Information Processing Systems, 2014.
[8] D. Tran, L.D. Bourdev, R. Fergus, L.T. M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: International Conference on Computer Vision, 2015, pp. 4489–4497.
[9] R. Baxter, N. Robertson, D. Lane, Human behavior recognition in data-scarce domains, Pattern Recognit. 48 (8) (2015) 2377–2393.
[10] H. Chen, G. Wang, J.-H. Xue, L. He, A novel hierarchical framework for human action recognition, Pattern Recognit. 55 (2016) 148–159.
[11] Y. Yi, M. Lin, Human action recognition with graph-based multiple-instance learning, Pattern Recognit. 53 (2016) 148–162.
[12] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, S.J. Maybank, Asymmetric 3d convolutional neural networks for action recognition, Pattern Recognit. 85 (2019) 1–12.
[13] T. Yu, C. Guo, L. Wang, H. Gu, S. Xiang, C. Pan, Joint spatial-temporal attention for action recognition, Pattern Recognit. Lett. 112 (2018) 226–233.
[14] J. Zhang, J. Yu, D. Tao, Local deep-feature alignment for unsupervised dimension reduction, IEEE Trans. Image Process. 27 (5) (2018) 2420–2432.
[15] L.L.C. Kasun, Y. Yang, G.-B. Huang, Z. Zhang, Dimension reduction with extreme learning machine, IEEE Trans. Image Process. 25 (8) (2016) 3906–3918.
[16] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L.V. Gool, Temporal segment networks: Towards good practices for deep action recognition, in: Europeon Conference on Computer Vision, 2016, pp. 20–36.
[17] C. Feichtenhofer, A. Pinz, R. Wildes, Spatiotemporal residual networks for video action recognition, Advances in Neural Information Processing Systems, 2016.
[18] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.
[19] H. Kwon, Y. Kim, J.S. Lee, M. Cho, First person action recognition via two-stream convnet with long-term fusion pooling, Pattern Recognit. Lett. 112 (2018) 161–167.
[20] H. Gao, L. Liu, V.D. Maaten, Densely connected convolutional networks, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2017.
[21] G.E. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, Advances in Neural Information Processing Systems Workshop, 2014.
[22] K. Soomro, A.R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, The Computing Research Repository, 2012.
[23] K. Soomro, A.R. Zamir, M. Shah, Hmdb: A large video database for human motion recognition, in: International Conference on Computer Vision, 2011, pp. 2556–2563.
[24] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: Europeon Conference on Computer Vision, 2006.
[25] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2008.
[26] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, in conjunction with the International Conference on Computer Vision, 2005.
[27] K. Derpanis, M. Sizintsev, K. Cannons, R. Wildes, Action spotting and recognition based on a spatiotemporal orientation analysis, IEEE Pattern Anal. Mach. Intell. 35 (3) (2013) 527–540.
[28] C. Feichtenhofer, A. Pinz, R. Wildes, Dynamically encoded actions based on spacetime saliency, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2015.
[29] A. Klaser, M. Marszaek, C. Schmid, A spatio-temporal descriptor based on 3d–gradients, in: British Machine Vision Conference, 2008.
[30] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1106–1114.
[31] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradientbased learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[32] V. Quoc, Y. Will, Y. Serena, A. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2011.
[33] G. Taylor, R. Fergus, Y. LeCun, C. Bregler, Convolutional learning of spatio-temporal features, in: Europeon Conference on Computer Vision, 2010.
[34] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.
[35] Y. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: Deep networks for video classification, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2015, pp. 4694–4702.
[36] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, 2016 arXiv:1604.04494.
[37] B. Mahasseni, S. Todorovic, Regularizing long short term memory with 3d human-skeleton sequences for action recognition, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2016.
[38] S. Sharma, R. Kiros, R. Salakhutdinov, Action recognition using visual attention, Advances in Neural Information Processing Systems workshop on Time Series, 2015.
[39] N. Ballas, L. Yao, C. Pal, A. Courville, Delving deeper into convolutional networks for learning video representations, in: International Conference on Learning Representations, 2016.
[40] Z. Li, E. Gavves, M. Jain, C.G. Snoek, Video lstm convolves, attends and flows for action recognition, 2016 arXiv:1607.01794.
[41] C. Bucila, R. Caruana, A. Niculescu-Mizil, Model compression: Making big, slow models practical, Knowledge Discovery and Data Mining, 2006.
[42] Z. Hu, X. Ma, Z. Liu, E.H. Hovy, E.P. Xing, Harnessing deep neural networks with logic rules, in: The Association for Computational Linguistics, August, 1, 2016, pp. 7–12.
[43] Z. Hu, Z. Yang, R. Salakhutdinov, E. Xing, Deep neural networks with massive learned knowledge, in: Conference on Empirical Methods in Natural Language Processing, Austin, Texas, USA, November 1–4, 2016, pp. 1670–1679.
[44] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: Hints for thin deep nets, in: International Conference on Learning Representations, 2015.
[45] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, in: International Conference on Learning Representations, 2017.
[46] Y. Chen, N. Wang, Z. Zhang, Accelerating deep metric learning via cross sample similarities transfer, The Association for the Advance of Artificial Intelligence, 2018.
[47] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.
[48] X. Wang, Y. Yan, P. Tang, X. Bai, W. Liu, Revisiting multiple instance neural networks, Pattern Recognit. 74 (2018) 15–24.
[49] J. Du, Y. Xu, Hierarchical deep neural network for multivariate regression, Pattern Recognit. 63 (2017) 149–157.
[50] J. Yu, B. Zhang, Z. Kuang, D. Lin, J. Fan, Iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning, IEEE Trans. Inf. Forensics Secur. 12 (5) (2017) 1005–1016.
[51] Y. Jiang, J. Liu, A.R. Zamir, I. Laptev, M. Piccardi, M. Shah, R. Sukthankar, Thumos challenge: action recognition with a large number of classes, 2013.
[52] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: a large-scale hierarchical image database, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2009, pp. 248–255.
[53] C. Feichtenhofer, A. Pinz, R. Wildes, Spatiotemporal multiplier networks for video action recognition, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2017.
[54] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice, Comput. Vision Image Understanding 150 (2016) 109–125.
[55] Z. Lan, M. Lin, X. Li, A.G. Hauptmann, B. Raj, Beyond gaussian pyramid: Multi-skip feature stacking for action recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 204–212.
[56] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep convolutional descriptors, in: Internaltional Conference on Computer Vision and Pattern Recogintion, 2015, pp. 4305–4314.
[57] W. Zhu, J. Hu, G. Sun, X. Cao, Y. Qiao, A key volume mining deep framework for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1991–1999.
[58] S. Ma, S.A. Bargal, J. Zhang, L. Sigal, S. Sclaroff, Do less and achieve more: training cnns for action recognition utilizing action images from the web, Pattern Recognit. 68 (2017) 334–345.
[59] X. Wang, A. Farhadi, A. Gupta, Actions transformations, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 2658–2667.
[60] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625–2634.
[61] R. Hou, C. Chen, M. Shah, Tube convolutional neural network (t-cnn) for action detection in videos, IEEE international conference on computer vision, 2017.

[62] L. Sun, K. Jia, D.-Y. Yeung, B.E. Shi, Human action recognition using factorized spatio-temporal convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4597–4605.
[63] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 4724–4733.

Wangli Hao is currently a Ph.D. candidate in the Center for Research on Intelligent Perception and Computing, Institute of Automation. She received her bachelor's degree in Shanxi Agricultural University in 2011. After that, She received her master's degree in Beijing Institute of Technology in 2014.

Zhaoxiang Zhang received his B.Sc. degree in Department of Electronic Science and Technology from University of Science and Technology of China, and the Ph.D. degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences in 2004 and 2009, respectively. From 2009 to 2015, he worked as a Lecturer, Associate Professor, and later the deputy director of Department of Computer Application Technology at the Beihang University. Since July 2015, Dr. Zhang has joined the National Laboratory of Pattern Recognition (NLPR) where he is currently a Professor. His major research interests include pattern recognition, computer vision, machine learning and bio-inspired visiual computing. He has published more than 150 papers in reputable conferences and journals. He has won the best paper awards in several conferences and championships in international competitions. He has served as the area chair, senior PC or PC of many international conferences like CVPR, ICCV, AAAI, IJCAI. He is the associate editor or guest associate editor of NeuroComputing, Pattern Recognition Letters, Cognitive Computation, IEEE Access and Frontiers of Computer Science.