Deep Bilinear Learning for RGB-D Action Recognition

Jian-Fang Hu[1], Wei-Shi Zheng[1,3,4][0000−0001−8327−0003] ⋆, Jiahui Pan[1], Jianhuang Lai[1], and Jianguo Zhang[2]

[1] Sun Yat-sen University, China
{hujf5,zhwshi,stsljh}@mail.sysu.edu.cn; panjh7@mail2.sysu.edu.cn;
[2] University of Dundee, United Kingdom
j.n.zhang@dundee.ac.uk;
[3] Key Laboratory of Machine Intelligence and Advanced Computing, MOE
[4] Inception Institute of Artificial Intelligence, United Arab Emirates

Abstract. In this paper, we focus on exploring modality-temporal mutual information for RGB-D action recognition. In order to learn time-varying information and multi-modal features jointly, we propose a novel deep bilinear learning framework. In the framework, we propose bilinear blocks that consist of two linear pooling layers for pooling the input cube features from both modality and temporal directions, separately. To capture rich modality-temporal information and facilitate our deep bilinear learning, a new action feature called modality-temporal cube is presented in a tensor structure for characterizing RGB-D actions from a comprehensive perspective. Our method is extensively tested on two public datasets with four different evaluation settings, and the results show that the proposed method outperforms the state-of-the-art approaches.

Keywords: Deep bilinear, RGB-D action, feature learning, Cube.

1 Introduction

Recognizing human actions based on low-cost depth camera has attracted increasing attention recently. Compared to RGB cameras, the Kinect, as one widely used depth camera, has many advantages. Firstly, it can capture depth maps, which was shown useful for geometric modeling [32]. Secondly, it can output 3D human poses (skeletons) in real-time, which also benefits action recognition [30].

Recent works have shown that the RGB, depth, and skeleton data captured by depth cameras can complement to each other for describing human actions; integrating them together can largely improve the system performance [37, 12, 39]. Specifically, in [37], the features extracted from different modalities and body parts are combined by a multi-kernel learning model. In [12, 28], features from various modalities are pooled together by explicitly mining the shared-specific components. However, the systems in these works only consider features from
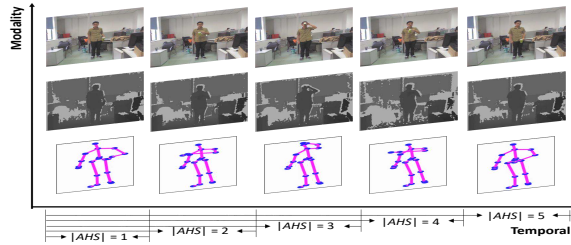

⋆ corresponding author

Fig. 1. Action snapshots with multi-modalities, showing actions can be recognized from sequences of different modalities and of different progress levels (the length of action history sequence (AHS, which will be discussed in detail in Section) 3.1.)

different modalities, all extracted from *full* action sequence. Relatively few works have explored the action context at different temporal levels, i.e, the time-varying information of sequences involving partial action executions.

Indeed, partial action executions in multi-modal sequences could contain informative action contexts from recognition perspective. Taking the action presented in Figure 1 for example, we can recognize that the person is drinking by observing any of the RGB, depth, or skeleton sequences. Meanwhile, the action can also be recognized by only observing the first 80% of the full sequence (i.e., $|AHS| = 4$), which means that sequences with partial action executions and of various modalities can be exploited in recognition. The use of time-varying information for action recognition could be traced back to the early work of motion history images (MHI) [2], where the history of motion is encoded in a single *static* image. Each MHI corresponds to one sequence at a certain progress level. However, few work has yet considered to deeply encode and learn the time-varying information together with the modalities. In this paper, we present a novel tensor-structured cube feature, and propose to learn time-varying information from multi-modal action history sequences for RGB-D action recognition.

The multi-modal sequences with temporal information can be regarded as a tensor, structured with two different dimensions (temporal and modality). Learning and pooling the tensor is a rather challenging task, due to the complexity of the arriving sequences, which are of varied progress levels and modalities. For the sequences at a certain progress level, since different modalities depict action from different perspectives, the features of varied modalities can complement to each other for describing actions context. While for a certain modality, sequences of various progress levels encode the temporal dynamics. And the time-varying information depicted in the sequences varies for different modalities. The time-varying information together with multi-modal features can give a comprehensive picture of the action, but how to learn the modality-temporal mutual information from highly structured sequence (tensor) remains a challenge.

In this paper, we address this challenge by proposing a novel deep bilinear framework, where a bilinear block consisting of two linear pooling layers (modality pooling layer and temporal pooling layer) is defined to pool the input tensor
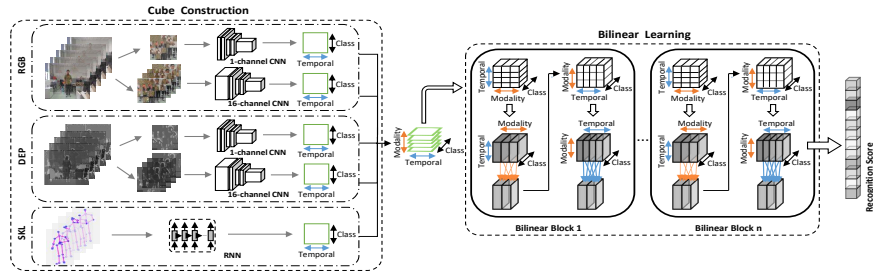
Fig. 2. Graphic illustration of our recognition system. Our system consists of two parts: cube feature construction and deep bilinear learning. The cube construction part is to extract multiple temporal feature maps for representing RGB-D actions. And the deep bilinear learning part is used to mine informative action representation for recognition.

along the modality and temporal directions, separately. In this way, the structures along the temporal and modal dimensions are both preserved. By stacking the proposed bilinear blocks and other network layers (e.g., Relu and softmax), we develop our deep bilinear model to jointly learn the action history and modality information in videos. Results have shown that learning modality-temporal mutual information is beneficial for the recognition of RGB-D actions.

Note that the use of bilinear pooling has also been explored in [9, 10] for pooling pair of features. However, their bilinear layer is defined as the outer product of two input features, which aims at pooling two vectors to a higher dimensional feature representation. These approaches are developed for pooling 1D vectors. In contrast, our objective is to integrate the input modality-temporal tensors from different dimensions, in order to preserve the tensor structures of the input. Our bilinear block is constructed based on the bilinear map, which learns the time-varying dynamics and multi-modal information in the sequences iteratively, and thus is more suitable for learning RGB-D sequences with complex tensor structures in the temporal and modality directions.

To encode rich modality-temporal information in the sequences and facilitate our deep bilinear learning, we further present a novel action descriptor called modality-temporal cube to characterize RGB-D actions from a comprehensive perspective. Our cube includes five feature maps, each of which is extracted from the sequences of various progress levels within a certain modality and describes actions from a certain perspective. Our experiments show that the proposed modality-temporal features fit the proposed deep bilinear model and can complement well to each other.

In summary, our contributions are: 1) a novel deep bilinear framework for learning multiple modality-temporal features; 2) a modality-temporal cube descriptor for characterizing RGB-D actions. Extensive experimental analysis and evaluations on two public benchmark RGB-D action sets, with four different evaluation settings, showing our method achieves state-of-the-art performances. A graphical illustration of our *system* is presented in Figure 2.
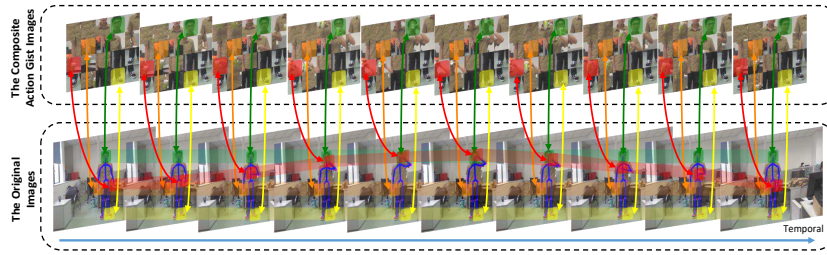
Fig. 3. Illustration of generating composite action GIST frames from original sequences.

2   Related work

In the following, we briefly review the approaches (depth or skeleton based and RGB-D based) for action recognition with Kinect, which are closely related to our work. We also outline the bilinear pooling techniques and the methods that learn multi-modal features and time-varying information for action recognition.

Depth or skeleton based action recognition. The geometric information depicted in depth sequences can be used to characterize action [24, 36, 42, 26, 18]. For instance, the histograms of oriented normal within each spatio-temporal depth cube was used to describe actions in [26, 42]. These methods mainly develop their systems based on the observed depth sequences. On the other hand, human action can also be characterized by the dynamics of human poses (or skeletons). The temporal dynamics of each skeleton joint [15, 40, 5, 33] and joint pairs [41, 25, 20, 43, 29] are explored for mining the structure motions depicted in the skeleton sequences. However, each of the modalities has its own insufficiency for characterizing complex actions involving objects and interactions. In comparison, our method explores the collaboration among different modalities, and thus the weakness of losing contextual information by only using depth or skeleton features can be overcome by working colloborately with RGB features.

RGB-D based action recognition. Recent works show that combining RGB, depth, and skeleton together can improve the system performance [37, 12, 39, 28, 19]. For instance, [13] proposed a joint learning framework to mine the structures shared and specified by different modal features. A deep shared-specific structure learning method is explored in [28]. Different from these works that choose to combine multi-modal features extracted from full sequences, in this paper, we formulate a deep learning approach to learn features from various modalities and progress levels. Thus the modality-temporal mutual structures are explored.

Bilinear pooling. Bilinear pooling has been introduced to combine features extracted by two CNN models [21, 9, 10]. In [9], for example, a deep architecture with bilinear pooling is developed for improving question answering. However in these works, bilinear pooling is defined as the outer product of two features in order to produce a higher dimensional feature. While in our work, bilinear

is defined as an operation block consisting of two linear operators pooling tensor features along modality and temporal dimensions, separately, which has the advantage of preserving tensor structures.

Multi-modal action recognition. Integrating multi-modal features can improve the recognition performance. A straightforward way to combine features is to directly concatenate them together [46, 31]. To mine more interactive information among multi-modal features, lots of methods are proposed to explicitly learn shared-specific structures among features [11, 13, 28]. However, these works do not explore the time-varying information among the multiple modal features extracted from sequences of different progress levels.

Time-varying information for action recognition. Studies show that explicitly capturing time-varying information in sequences is beneficial. Intuitively, the time-varying information can be captured by a non-parametric model like mean or max pooling [16] and Fourier transform [13] etc. Learning time-varying information by data-driven approaches [7, 8, 35] can generalize better to unseen sequences. For example, [7] used a ranking machine to encode the dynamics among the sequential features. Note that the TSN [38] also intends to learn time-varying information within sequences of varies modalities. However, they modeled the time-varying and modality-varying information isolately. The time-varying information mined from each modality is empirically summarized, which makes their method less applicable for modelling temporal-modality mutual information. In contrast, we develop a flexible learning framework for learning the dynamics among sequences of various modalities and temporal lengths jointly.

3 Approach

We aim to explore the time-varying and modality-varying information for RGB-D action recognition by proposing a novel deep bilinear framework, which aims to integrate modality-temporal cubes in the modality and temporal directions. We also present a cube descriptor for characterizing RGB-D actions.

3.1 Modality-temporal Cube Construction

Here, we describe how to construct our modality-temporal cube for representing RGB-D actions. Our cube includes temporal feature maps extracted from the sequences of various progress levels within a certain modality (skeleton, RGB or depth), each of which characterizes actions from a certain perspective.

Action History Sequence. For extracting temporal features, we uniformly divide each sequence into D segments and consider the sequence including the first d segments as an *action history sequence* (AHS) with length d ($|AHS| = d$). Therefore, we have a total of D AHSs, whose lengths range from 1 to D. Then, for each sequence of skeleton, RGB, or depth, we extract temporal features from the corresponding AHSs, which forms the base to capture time-varying information.

Skeleton Temporal Feature Map. We employ a sequence-sequence RNN to extract temporal features from each skeleton sequence, where the AHSs are

encoded by the dynamic skeleton descriptor (DS) [13]. Thus, the dependencies among the DS features of consecutive AHSs are modeled. Then, we use the outputs of RNN as our feature map, which can capture some dynamic skeleton information depicted in each sequence.

RGB and Depth Temporal Feature Maps. Inspired by [13], where the visual features extracted from local image patches around each skeleton joint are used to represent human action, we also consider extracting our temporal feature maps in a similar way. Here, for each RGB/D image frame, we collect the local image patches around each skeleton joint, and tile them to compose a new image, which we termed as action *gist* image, a compact representation of the action frame as illustrated in Figure 3. Therefore, an action gist sequence are formed by pooling its GIST frames sequentially. Noted that local patches corresponding to the same (tracked) skeleton joint are tiled at the same spatial location in the frame, but across time, forming a trajectory-based patch sequence in the temporal dimension. There are two merits of using such a composition: 1) it enables efficient training of trajectory-based CNN as we don't need to train a CNN for each trajectory-based patch sequence; and 2) it captures the dynamics of patch appearances along each trajectory. In Figure 3, we have presented some examples about the composite action GIST frames. As can be seen, the gist image frames condense most of the action context and automatically remove the irrelated information, such as background. Patches at the same spatial location correspond to a long-term trajectory of a joint. In this end, our work could be among the family of trajectory-based action recognition [34].

Then, we construct our RGB and depth temporal feature maps by extracting K-channel CNN[5] descriptors from all the composite action gist AHSs, respectively. To train K-channel CNN, we selected K ordered action GIST frames for each training sequence. Specifically, the temporal location of the u-th selected frame is given by $max(1, 1 + (u - 1)\frac{ls}{K} + \delta)$, where ls indicates the length of sequence and perturbation δ is a random integer obeying uniform distribution $U(-\frac{ls}{2K}, \frac{ls}{2K})$. In our experiments, two different settings ($K = 1$ and $K = 16$) are used. The feature map extracted from $K = 1$ can capture static appearance information, while the map from $K = 16$ characterizes dynamic appearance.

Feature Cube Construction. Finally, we concatenate all the feature maps along the modality dimension to construct the modality-temporal cube, whose size is modality number×AHS number×feature dimension. In total, our cube descriptor contains five temporal feature maps, with two from RGB AHSs (1-channel CNN and 16-channel CNN), two from depth AHSs ((1-channel CNN and 16-channel CNN), and one from the skeleton AHSs (RNN), each of which characterizes actions at different AHS lengths from a specific modality. The combination of them can form a comprehensive action representation.

Note that for constructing the temporal feature for the AHS of a specific modality and temporal length, we use the output of the final layer of CNN (or RNN for skeleton AHSs), whose size is the same as the number of action

---

[5] The input of K-channel CNN is K gray images concatenated along the channel dimension. Thus, it is a CNN whose input size is $224 \times 224 \times K$.
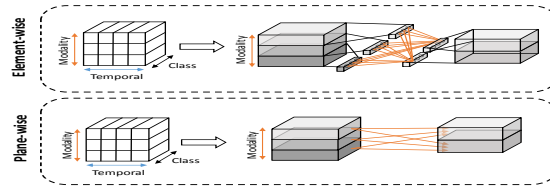
Fig. 4. Pooling by element-wise fully connected vs. plane-wise fully connected layer.

classes. Those features can be considered as soft classification scores (i.e., before the use of softmax operator). Thus, the third dimension of our cube encodes the classification information, and the elements along this dimension are highly related with each other. We call this feature dimension as the *class* dimension.

3.2 Deep Bilinear Learning

Our cube descriptor includes multiple temporal features extracted from RGB-D AHSs, making most of the existing multi-modal feature learning methods not applicable to learn an informative action representation. As each element in the (cube) class dimension corresponds to the confidence of assigning the given sample to a certain action class, pooling the confidences of different classes does not make much sense. Moreover, our experimental results in Table 5 confirm that merging elements of different classes is not the best for our framework. In the following, we introduce a novel deep learning framework to pool the modality and temporal information, while keeping the class dimension unchanged. We call our framework *deep bilinear* as it is inspired by the formulation of bilinear map.
Bilinear Map Revisited. In mathematics, a bilinear map is a function combining elements of two vector spaces to yield an element of a third vector space. The formulation of a widely used bilinear function in the community is

$$f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y} \tag{1}$$

where $\boldsymbol{A} \in R^{m \times n}$, $\boldsymbol{x} \in R^m$, and $\boldsymbol{y} \in R^n$. As can be seen, $f(\boldsymbol{x}, \boldsymbol{y})$ is linear with respect to each of the variables $\boldsymbol{x}$ and $\boldsymbol{y}$.

It is straightforward to extend the above formulation in the matrix form as

$$f(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{X}^T \boldsymbol{A} \boldsymbol{Y} \tag{2}$$

where $\boldsymbol{A} \in R^{m \times n}$, $\boldsymbol{X} \in R^{m \times p}$, and $\boldsymbol{Y} \in R^{n \times q}$. This formula can be considered as a combination of two linear operators. The first operator $\boldsymbol{L} = \boldsymbol{X}^T \boldsymbol{A}$ is to combine the rows of $\boldsymbol{A}$ using the weights indicated by the columns of $\boldsymbol{X}$. It pools the rows of the input matrix, while holding the column dimension constant. We call it *row-pooling operator*. And the second operator $\boldsymbol{LY}$ (named *column-pooling operator*) is to calculate the weighted summation of all the columns in the latent matrix $\boldsymbol{L}$, where the combining weights are indicated by the rows of

Y. It is used to pool the columns of L. The combination of the row-pooling and column-pooling transforms the $m \times n$-sized A to a matrix of $p \times q$.

Bilinear Block. Given a modality-temporal cube, here we would define a block, named *bilinear block*, to pool it in the modality and temporal dimensions, separately, based on the bilinear map (2). Therefore, the tensor structures along the modality and temporal dimensions are preserved during feature pooling. Note that the block would keep the *class* dimension constant. Our bilinear block is consisted of two neural layers (i.e., temporal pooling layer and modality pooling layer), each of which corresponds to one operator in the bilinear function.

Modality pooling layer. This layer is defined to pool the input cube in the modality dimension. We formulate it as a plane-wise linear combination problem:

$$L(:,:,c) = X^T A(:,:,c), c = 1, 2, ..., C \tag{3}$$

where $X \in R^{M_A \times M_L}$ is the model parameter to be learned, where M_A and M_L are the modality dimension of cube A and L. Specially, M_L is a parameter to be specified by the user. $A \in R^{M_A \times T \times C}$ is the input cube and L is the output cube, whose size is $M_L \times T \times C$. The layer defined by Equation (3) pools the modality dimension from M_A to M_L. Let's denote the layer as f_M for simplification.

It is worth noting that the modality pooling layer (3) can be rewritten as

$$L(m_L,:,:) = \sum_{m_A=1,2,...,M_A} X(m_A, m_L) A(m_A,:,:), m_L = 1, 2, ..., M_L \tag{4}$$

which means that elements corresponding to the same modality are weighted by the same parameter. That is, the cube is pooled in a plane-wise manner. An alternative way is to pool it in an element-wise manner, where each element is weighted by a specific parameter, as illustrated in Figure 4. However, this would introduce a large number of learnable parameters, making the model easily fall into over-fitting. We will demonstrate it in the experiment section (5).

Temporal pooling layer. The temporal pooling layer is defined to pool the input 3D cube in the temporal dimension. Specifically, it can be formulated as

$$Z(:,:,c) = L(:,:,c)Y, c = 1, 2, ..., C \tag{5}$$

here, Z and Y indicate the output cube and the pooling parameters, respectively.

We would like to point out that the temporal pooling layer can be equivalently calculated using the modality pooling layer if we permute the temporal dimension and modality dimension of the input cubes. In the following, we use f_T to indicate the temporal pooling layer. To improve the generalization capability, we additionally constrain the model parameters X (Y), corresponding to each layer in the block, by L_2-norm and L_1-norm constraint. The L_1-norm is employed to penalize non-zero elements in X (Y), which could result in a sparse solution. The L_2-norm serves as a decay term.

Then the bilinear block can be defined by $b = f_T \circ f_M(A)$. Here, we construct our bilinear block based on the modality pooling and temporal pooling layers, pooling the cube from one dimension to another, separately.
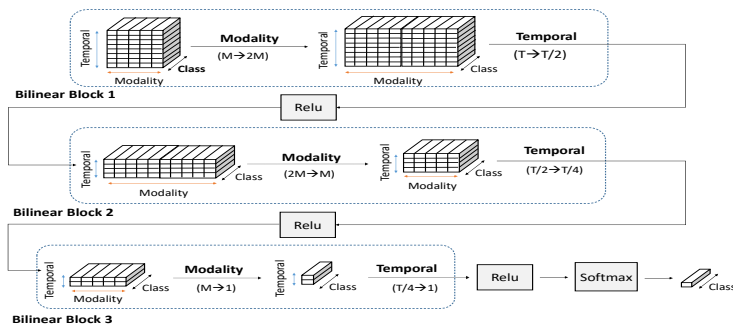
Fig. 5. Graphic illustration of the employed deep architecture.

Deep Bilinear Architecture. Given a set of $M \times T \times C$-sized modality-temporal cubes, our goal is to learn an underlying mapping f, which merges all the cube elements into a robust representation $\boldsymbol{y} \in R^C$. In other word, the objective is to find a mapping that pools the modality dimension and temporal dimension of the input cube to 1. In this paper, we define the mapping f as a stack of bilinear blocks, Relu, and softmax operators, i.e., $f = g_1 \circ g_2 \circ ...g_n...(\bullet)$, where g_n refers to one of the above operators or bilinear block.

The form of our deep bilinear architecture is flexible. Experiments in this paper involve a deep architecture with three bilinear blocks, three Relu layers and a softmax layer, while more layers are possible. In the architecture, each bilinear block is followed by a Relu layer to map the outputs of the block non-linearly. A graphic illustration for the employed deep architecture can be found in Figure 5. Please refer to the experiment section for more details.

Optimization. We optimize our deep bilinear by stochastic gradient descent (SGD) with momentum, where the gradients are determined by back propagation algorithm. We use the logistic loss as our loss function. For the gradient of L_1-norm of $\boldsymbol{X}$ ($\boldsymbol{Y}$), we use the generalized gradient $\boldsymbol{X}./|\boldsymbol{X}|$ ($\boldsymbol{Y}./|\boldsymbol{Y}|$) for simplicity.

4   Experiment

We evaluated our methods on two public benchmark 3D action datasets: NTU RGB+D Dataset [22] and SYSU 3D HOI dataset [14], with two different evaluation protocols employed in each set. In the following, we will briefly introduce the implementation details and then describe our experimental results.

4.1   Implementation Details

Following the observation in [13], we extract the 64×64 patches around the skeleton joints to form our composite action GIST frames[6]. For extracting temporal feature maps from RGB and Depth videos on the NTU RGB+D set, we

---

[6] The gist images are linearly resized to 224×224.

Table 1. Recognition results (accuracies) on the NTU RGB+D set. '—' indicates that the performance is not reported in the literature.

Method	Data Used	Cross-subject	Cross-view
Skeletal Quads [6]	SKL	38.6%	41.4%
Dynamic Skeletons [12]	SKL	60.2%	65.2%
Part-aware LSTM [27]	SKL	62.9%	70.3%
ST-LSTM [22]	SKL	69.2%	77.7%
GCA-LSTM network [23]	SKL	74.4%	82.8%
Deep multi modal [28]	RGB+DEP+SKL	74.9%	—
MTLN [17]	SKL	79.6%	84.8%
View-adaption LSTM [44]	SKL	79.4%	87.6%
Pose-attention [1]	RGB+SKL	82.5%	88.6%
Deep Bilinear	RGB+DEP	79.2%	81.1%
	RGB+SKL	83.0%	87.1%
	DEP+SKL	83.3%	89.5%
	RGB+DEP+SKL	**85.4%**	**90.7%**

trained a set of K-channel VGG-16 networks without pre-training on other auxiliary datasets[7], where we set the momentum factor and dropout rate as 0.9 and 0.7, respectively. While for the SYSU 3D HOI dataset, since we do not have enough data to train CNN, we chose to finetune the models trained on the NTU RGB+D set. For the training of RNN on both sets, we used the back propagation through time (BPTT) algorithm with momentum for optimization, where the momentum rate was set as 0.9. The neuron number in the hidden layer of RNN was set as 256. To speed up the optimization of RNN, we used PCA to reduce the dimension of the extracted DS features, where 98% of variance is retained.

In the following experiments, our deep bilinear learning model is defined as a stack of three bilinear blocks, three Relu layers and one softmax layer, unless stated otherwise. The detailed architecture is modality pooling layer M⟶2M, temporal pooling layer T⟶T/2, modality pooling layer 2M⟶M, temporal pooling layer T/2⟶T/4, Relu, modality pooling layer M⟶1, temporal pooling layer T/4⟶1, Relu, softmax , which is illustrated in Figure 5. Here modality pooling layer 2M⟶M means the layer pools the cube in the modality dimension from 2M to M. T, C, M indicate the temporal length, class number, and modality number, respectively. We empirically found that upscaling the modality dimension can produce better recognition results in our experiments. It might be because that features of different modalities have large variations and upscaling modality dimension can produce meta-modal features with better expressive power, which is in line with the basic idea of developing kernel tricks. The model parameters are initialized by an altered xavier algorithm, where the random weights are produced by an uniformly distribution rather than a Gaussian distribution. We experimentally find that initializing the network in this way can significantly reduce the time of training. Temporal feature maps extracted from AHSs containing 70%-100% of the full sequence (i.e., $|AHS|$=7, 8, 9, 10) are

---

[7] Indeed, we do not observe a significant improvement in the recognition performance by pre-training the network on the imageNet set.

Table 2. Recognition results (accuracies) on the SYSU 3D HOI set. '—' indicates that the performance is not reported in the literature.

Method	Data Used	setting-1	setting-2
HON4D [13, 26]	DEP	73.4%	79.2%
HFM [13, 4]	RGB+DEP+SKL	75%	76.7%
ST-LSTM [22]	SKL	76.5%	—
View-adaption LSTM [44]	SKL	76.9%	77.5%
MPCCA [13, 3]	RGB+DEP+SKL	76.3%	80.7%
MTDA [13, 45]	RGB+DEP+SKL	79.2%	84.2%
JOULE [13]	RGB+DEP+SKL	79.6%	84.9%
Deep Bilinear	RGB+DEP	77.2%	83.1%
	RGB+SKL	81.5%	86.2%
	DEP+SKL	82.6%	84.8%
	RGB+DEP+SKL	**84.8**%	**88.9%**

used to construct the cube descriptor in most of the experiments. The learning rate is initialized as 10^{-3} and it would drop to 10^{-4} after several iterations.

4.2   NTU RGB+D Dataset

The NTU RGB+D dataset was specifically collected for the researches of large scale RGB-D human action recognition. For collecting this set, 40 subjects were asked to perform 60 different actions and the complete action executions were captured from three different views using a Kinect v2. In total, it contains more than 56K action samples for both training and testing. Compared to most of the existing dataset, this set is very challenging and larger in terms of the number of action classes, views, and samples with large intra-class variations [13, 37]. For experiment, we follow exactly the same evaluation settings specified in [22], where two different training-testing splits (i.e. cross-subject and cross-view) are used to evaluate the recognition performances. In the cross-subject setting, the sequences performed by 20 subjects are used to train, and the rest to test. While in the cross-view setting, samples for two views (camera 2 and camera 3) are used as training set, and the other samples form the testing set.

The comparison results are presented in Table 1. As shown, our approach with deep bilinear learning obtains the best results on this set and outperforms the state-of-the-art approaches, such as MTLN [17] and View-adaption LSTM model [44], by a large margin (e.g., $\geq 6\%$ for the cross subject setting). In detail, our method obtains an accuracy of 85.4% and 90.7% for the cross-subject and cross-view setting, respectively. We can observe that even for the cross-view setting, our model can still perform better than all the other competitors, and in particular outperforms the view adaption model [44] by 3.1%, which was specifically designed for recognizing actions across different views. It is interesting to note that our bilinear framework performs better than the model developed in [28] (85.4% vs. 74.9%), which also learns features extracted from RGB, depth, and skeleton by a deep model, however only using full sequences. This demonstrates the efficacy of our bilinear framework, which aims at exploring AHS with partial action executions and of different modalities for action recognition.

We can also observe that even using the temporal feature maps extracted from two of the RGB, depth, and skeleton data, we can still obtain a good performance, which is comparable to the state-of-the-art models, e.g., Pose-attention network. This means that explicitly mining some informative modality-temporal structures with our deep bilinear model is beneficial for recognition. As expected, the performance is largely improved when we fuse all the features together using the proposed deep bilinear learning algorithm. This also indicates that the temporal feature maps extracted from different modality sequences can complement well to each other for obtaining a comprehensive action representation.

4.3   SYSU 3D HOI set

The SYSU 3D HOI set was collected for studying complex actions with human-object interactions. This set contains 480 samples from 6 pairs of interaction actions including *playing with a cell-phone* and *calling with a cell-phone*, *mopping* and *sweeping* etc. This set is challenging because each pair of the considered interactions contains similar object contexts and interactive motions. For experiments, we employ the two evaluation criterions defined in [14] to test. In the first setting (named setting-1), for each action class, half of the samples are used for training and the rest for testing. The second setting (named setting-2) is a cross-subject setting, where sequences performed by half of the subjects are used to train the model parameters and the rest to test. For each setting, the mean accuracies obtained by 30 random training-testing splits are reported.

We report the results in Table 2. As can be seen, in both settings, our deep bilinear model outperforms the state-of-the-art model JOULE [13], which aims to learn action representation from the full sequences of different modalities. Especially for the setting-1, our method has a performance gain of 4.8%. This indicates that explicitly exploring time-varying information depicted in multiple modality sequences is beneficial for RGB-D action recognition . The same as that on NTU RGB+D set, fusing the multiple modality-temporal cube descriptors can obtain much better performances, which illustrates that the our deep bilinear model can learn a comprehensive action representation from the cubes for characterizing human actions. We can also observe that the RGB-D based models (JOULE [13] and our deep bilinear model) obtain better results than the single modality based methods (e.g. View-adaption LSTM [44], ST-LSTM [22], and HON4D [26]). This is as expected as only using depth or skeleton data is intrinsically limited in overcoming the ambiguity caused by appearance changes, occlusion, cluttered background, etc.

5   Analysis in Depth

Here, we provide more discussions and analysis on the proposed deep bilinear learning method. All the following conclusions are obtained based on the experiments on NTU RGB+D dataset with the challenging cross-subject setting.

Table 3. Evaluations on with vs. without temporal modelling.

	RGB	DEP	Skeleton	RGB+DEP+SKL
without	72.5%	73.1%	69.8%	83.5%
with	75.3%	75.9%	72.5%	85.4%

Table 4. Evaluations on the lengths of AHSs.

AHS lengths	≥ 1	≥ 3	≥ 5	≥ 7	≥ 9
Accuracy	84.1%	84.1%	84.3%	85.4%	84.3%

Evaluations on the temporal modelling. Our deep bilinear model learns dynamics from modality-temporal cubes. Here, we study the influence of the temporal dimension by only using the features corresponding to full sequences. The detailed results are presented in Table 3. As shown, with temporal dynamic modelling, we can see a valuable improvement (about 1.5-3% in the term of accuracy), which demonstrates the efficacy of learning time-varying information among AHSs of varied lengths for action recognition.

Here, we further study the influence of the lengths of the AHSs. We test on the AHSs whose lengths are larger than or equal to $1, 3, 5, 7, 9$, respectively. The results are presented in Table 4. We can observe that our system obtains the best result when the length is larger than or equal to 7. The accuracy would drop when the length goes smaller. This is because the AHSs with small length do not contain enough action context for characterizing actions. Introducing short AHSs could add more noise to the learning.

Comparison with other fusion and bilinear schemes. Here, we compare our bilinear learning framework with other fusion and bilinear schemes. Specifically, we test different settings in which cube are pooled by max pooling (max), mean pooling (mean), linear SVM, and multi-modal compact bilinear (MCB [9]) models. We also replace the plane-wise connected pooling in our bilinear block (denoted by Ours in Table 5) by the element-wise FCN (see Figure 4 for details) and compare their performances. The comparison results are presented in Table 5. As can be seen, our model offers distinct advantages over the hard-coded non-learning fusion methods (e.g., max and mean). This is because each layer of block in our model is specifically driven by either modality or temporal variate. Thus our bilinear model offers learning capability towards better fusion. While these hard-coded methods lack this key point. By examining the results obtained by the data-driven fusion schemes (e.g., FCN, linear-SVM, MCB and multi-kernel

Table 5. Comparison with other fusion schemes, which used our feature netowrks.

Method	max	mean	Linear SVM	FCN	MCB [9]	MKL	Ours
Parameter No.	0	0	72K	6.7M	16K	$\approx$ 72K	115
Accuracy	77.5%	83.0%	83.5%	76.3%	84%	84.1%	85.4%

Table 6. Evaluations on the number of bilinear blocks used in deep bilinear model.

Block number	1	2	3	4	5
Accuracy	83.8%	84.4%	85.4%	85.1%	84.9%

learning (MKL)), we can see that data-driven fusion can achieve better results than the hard-coded ones. The best result among them is achieved by MKL, with an accuracy of 84.3%, which outperforms all other methods in the table except ours. It is also noted that if we use element-wise FCN to pool cube descriptor instead of the plane-wise one, the performance decreases. This is as expected, as FCN has a large number of parameters to be learned, which makes the model easily fall into over-fitting. And the more parameters the model has, the worse performance is observed. Our method also outperforms the MCB [9] by 1.4%, which pools the features by an out-product bilinear operator without exactly considering the tensor structures in different dimensions. This demonstrates that learning temporal-modality mutual information in an iterative manner with our bilinear model can help to enhance recognition performance.

Effect of bilinear depth and pooling order. Our deep bilinear is constructed by stacking a set of bilinear blocks and other network layers. Here, we evaluate the influence of the number of bilinear blocks (depth). The results are listed in Table 6. It could be observed that when the number of blocks is small, increasing the depth will increase the performance (e.g., 85.4% vs. 83.8%); when the number gets larger (e.g., larger than 3), performance tends to saturate, being insensitive to the increase of depth. Our method is also not sensitive to the order of fusion. For example, if we fuse the temporal dimension first and then fuse over modality in each bilinear block, the recognition accuracy drops slightly (85.0% vs. 85.4%).

6   Conclusion

We present a novel deep bilinear learning framework to learn modality-temporal information (i.e., time-varying information across varies modalities) for RGB-D action recognition. In the framework, a bilinear block consisting of two linear pooling layers is constructed to extract the mutual information from modality and temporal directions, respectively. Furthermore, we present a new action feature representation to encode the action context in a tensor structure, named modality-temporal cube. Extensive experiments have been reported to demonstrate the efficacy of the proposed framework.

Acknowledgment

References

1. Baradel, F., Wolf, C., Mille, J.: Human action recognition: Pose-based attention draws focus to hands. In: International Conference on Computer Vision Workshop (2017)
2. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Transactions on pattern analysis and machine intelligence $\mathbf{23}(3)$, 257–267 (2001)
3. Cai, Z., Wang, L., Peng, X., Qiao, Y.: Multi-view super vector for action recognition. In: International Conference on Computer Vision and Pattern Recognition. pp. 596–603 (2014)
4. Cao, L., Luo, J., Liang, F., Huang, T.S.: Heterogeneous feature machines for visual recognition. In: International Conference on Computer Vision. pp. 1095–1102 (2009)
5. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: International Conference on Computer Vision and Pattern Recognition. pp. 1110–1118 (2015)
6. Evangelidis, G., Singh, G., Horaud, R.: Skeletal quads: Human action recognition using joint quadruples. In: International Conference on Pattern Recognition. pp. 4513–4518 (2014)
7. Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., Tuytelaars, T.: Rank pooling for action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence $\mathbf{39}(4)$, 773–787 (2017)
8. Fernando, B., Gould, S.: Learning end-to-end video classification with rank-pooling. In: International Conference on Machine Learning. pp. 1187–1196 (2016)
9. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. arXiv preprint arXiv:1606.01847 (2016)
10. Gao, Y., Beijbom, O., Zhang, N., Darrell, T.: Compact bilinear pooling. In: International Conference on Computer Vision and Pattern Recognition. pp. 317–326 (2016)
11. Gu, Q., Zhou, J.: Learning the shared subspace for multi-task clustering and transductive transfer classification. In: International Conference on Data Mining. pp. 159–168 (2009)
12. Hu, J.F., Zheng, W.S., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. In: International Conference on Computer Vision and Pattern Recognition. pp. 5344–5352 (2015)
13. Hu, J.F., Zheng, W.S., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. IEEE transactions on pattern analysis and machine intelligence $\mathbf{39}(11)$, 2186–2200 (2017)
14. Hu, J.F., Zheng, W.S., Ma, L., Wang, G., Lai, J.: Real-time rgb-d activity prediction by soft regression. In: European Conference on Computer Vision. pp. 280–296 (2016)
15. Hussein, M.E., Torki, M., Gowayyed, M.A., El-Saban, M.: Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In: International Joint Conferences on Artificial Intelligence. vol. 13, pp. 2466–2472 (2013)
16. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: International Conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)

17. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. arXiv preprint arXiv:1703.03492 (2017)
18. Klaser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: British Machine Vision Conference. pp. 275–1 (2008)
19. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. International Journal of Robotics Research **32**(8), 951–970 (2013)
20. Lillo, I., Soto, A., Niebles, J.C.: Discriminative hierarchical modeling of spatio-temporally composable human activities. In: International Conference on Computer Vision and Pattern Recognition. pp. 812–819 (2014)
21. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: IEEE International Conference on Computer Vision. pp. 1449–1457 (2015)
22. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. In: European Conference on Computer Vision, pp. 816–833 (2016)
23. Liu, J., Wang, G., Hu, P., Duan, L.Y., Kot, A.C.: Global context-aware attention lstm networks for 3d action recognition. In: International Conference on Computer Vision and Pattern Recognition. pp. 1647–1656 (2017)
24. Lu, C., Jia, J., Tang, C.K.: Range-sample depth feature for action recognition. In: International Conference on Computer Vision and Pattern Recognition. pp. 772–779 (2014)
25. Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R.: Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. Journal of Visual Communication and Image Representation **25**(1), 24–38 (2014)
26. Oreifej, O., Liu, Z.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: International Conference on Computer Vision and Pattern Recognition. pp. 716–723 (2013)
27. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. arXiv preprint arXiv:1604.02808 (2016)
28. Shahroudy, A., Ng, T.T., Gong, Y., Wang, G.: Deep multimodal feature analysis for action recognition in rgb+ d videos. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(3), 305–317 (2017)
29. Shahroudy, A., Ng, T.T., Yang, Q., Wang, G.: Multimodal multipart learning for action recognition in depth videos. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(10), 2123–2129 (2016)
30. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56**(1), 116–124 (2013)
31. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. pp. 568–576 (2014)
32. Smisek, J., Jancosek, M., Pajdla, T.: 3d with kinect. In: Consumer depth cameras for computer vision, pp. 3–25 (2013)
33. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 588–595 (2014)

34. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision **103**(1), 60–79 (2013)
35. Wang, H., Wang, L.: Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. arXiv preprint arXiv:1704.02581 (2017)
36. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3d action recognition with random occupancy patterns. In: European Conference on Computer Vision, pp. 872–885 (2012)
37. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Learning actionlet ensemble for 3d human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(11), 379–385 (2013)
38. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision (2016)
39. Wei, P., Zhao, Y., Zheng, N., Zhu, S.C.: Modeling 4d human-object interactions for event and object recognition. In: International Conference on Computer Vision. pp. 3272–3279 (2013)
40. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3d joints. In: International Conference on Computer Vision and Pattern Recognition. pp. 20–27 (2012)
41. Yang, X., Tian, Y.: Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: International Conference on Computer Vision and Pattern Recognition. pp. 14–19 (2012)
42. Yang, X., Tian, Y.: Super normal vector for activity recognition using depth sequences. In: International Conference on Computer Vision and Pattern Recognition. pp. 804–811 (2014)
43. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: International Conference on Computer Vision. pp. 2752–2759 (2013)
44. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. arXiv preprint arXiv:1703.08274 (2017)
45. Zhang, Y., Yeung, D.Y.: Multi-task learning in heterogeneous feature spaces. In: Conference on Artificial Intelligence (2011)
46. Zhu, Y., Chen, W., Guo, G.: Fusing spatiotemporal features and joints for 3d action recognition. In: International Conference on Computer Vision and Pattern Recognition. pp. 486–491 (2013)