Front. Comput. Sci., 2017, 11(2): 219–229
DOI 10.1007/s11704-016-6066-5

RESEARCH ARTICLE

Attribute-based supervised deep learning model for action recognition

Kai CHEN[1], **Guiguang DING** (✉)[1], **Jungong HAN**[2]

1 School of Software, Tsinghua University, Beijing 100084, China
2 Department of Computer Science, Northumbria University, Newcastle NE1 8ST, UK

Abstract Deep learning has been the most popular feature learning method used for a variety of computer vision applications in the past 3 years. Not surprisingly, this technique, especially the convolutional neural networks (ConvNets) structure, is exploited to identify the human actions, achieving great success. Most algorithms in existence directly adopt the basic ConvNets structure, which works pretty well in the ideal situation, e.g., under stable lighting conditions. However, its performance degrades significantly when the intra-variation in relation to image appearance occurs within the same category. To solve this problem, we propose a new method, integrating the semantically meaningful attributes into deep learning's hierarchical structure. Basically, the idea is to add simple yet effective attributes to the category level of ConvNets such that the attribute information is able to drive the learning procedure. The experimental results based on three popular action recognition databases show that the embedding of auxiliary multiple attributes into the deep learning framework improves the classification accuracy significantly.

Keywords action recognition, convolutional neural network, attribute

1 Introduction

An action is a sequence of human body movements, indicating the person's intentions and thoughts. From the perspective of computer vision, the recognition of an action is to parse the video sequence in order to learn about the action, and in turn, the learned knowledge is employed to identify similar actions when they appear again. As a key component in human behavior analysis and understanding, automatic action recognition facilitates many applications such as human computer interaction, video surveillance [1–3] and video analysis & retrieval [4–6], which is shown in Fig. 1.

(a) (b) (c)

Fig. 1 Application Exemplars where action recognition can be used. (a) Interaction; (b) surveillance; (c) video analysis

Currently, the popular methods for action recognition can be generally divided into two categories. Traditional methods employ efficient handcrafted feature descriptors to represent an action, in which the appearance and texture information are commonly encoded. The representatives include histogram of oriented gradient (HOG) [7], histogram of optical flow (HOF) [8] and motion boundary histogram (MBH) [9]. Alternatively, moving trajectory [10] is also well investigated and the descriptors based on it achieve success in some applications. On the other hand, emerging techniques try to adapt learning-based feature representations to human action recognition area. For example, bag of words (BoW) [11] model and sparse coding [12] have been extensively adopted, which are capable of handling the diversity of local features. The information from multi-view depth images

Received February 1, 2016; accepted December 8, 2016

E-mail: dinggg@tsinghua.edu.cn

has also been focused [13]. Recently, deep learning methods, such as data-driven convolutional neural network [14], are used to obtain intrinsic representation from the training samples, which have shown compelling preliminary results. Additionally, some attempts combine the temporal handcrafted features and deep learning methods [15]. In contrast to the handcrafted feature descriptors, learning based feature representations tend to be domain agnostic, and are able to learn additional feature bases that cannot be represented through any of the handcrafted features.

Although deep learning framework has been employed in human action recognition, the performance of the algorithm is still far from satisfactory, especially when dealing with the practical situations. A crucial problem is that there is a huge "*gap*" between the deep learned features and the semantic actions due to the fact that such methods usually learn features from the raw image. With respect to the action recognition, the semantic patterns such as temporal changes of the feature may help to describe action. To this end, one potential idea is allowing the semantic information to supervise the feature learning in the deep learning framework.

Inspired by the above analysis, in this paper, a human action recognition algorithm based on deep learning framework is proposed, into which we successfully integrate the semantic-level attribute information. Our work differs from the existing work in two aspects.

1) We propose a novel deep learning model supervised by the attribute information, namely Attribute-based Supervised Deep Learning Model. With the semantic-level attribute integrated, the learned features are more robust to the changes of image appearance that arise within the same category.

2) We propose two simple yet effective attributes describing human action at the semantic level, one of which indicates the background information, and the other one describes the video super category. These attributes are successfully integrated into the category level of ConvNets. On the one hand, the involvement of such attributes in CNN framework makes the whole system more efficient, reducing the annotation work required in the training procedure. On the other hand, the way of embedding attribute information into the inflexible deep learning framework provides an implementation example, which may inspire the research attempting to modify the ConvNets model.

2 Related work

As our work is a sort of improved work of deep learning based feature representation, we focus on discussing feature extraction by means of learning techniques.

Learning-based feature representation has been extensively adopted to encode local features. BoW [11] model is a popular tool to quantify various local features into a unified feature space. In computer vision, BoW model usually treats local image features as words and generates a sparse histogram over the whole feature space by counting the occurrence of features. It provides a way to decrease the descriptors' amount, and to help learn and generate an image descriptor using a long and sparse vector. Despite its excellent feature representation capability, BoW model has some limitations, in which one of the disadvantages is the neglect of the spatial relationships among image patches. Sparse coding [12] is another common unsupervised method to find valid representations that capture higher-level features, given unlabeled input data. At its training stage, sparse coding method tries to learn a small number of bases to represent the input data, which minimize the object function. Then the bases can be used to encode the data, thus obtaining the feature representation. Using a sparse matrix to store the features can save a lot of space with the cost of sacrificing little information in an acceptable degree. Principal components analysis (PCA) [16] is a technology that simplifies and extracts the feature representation. PCA methods obtain the principal components and their feature weights of input data, through performing characteristics decomposition for covariance matrix. Finally it contains the features which have maximal contributions to the data, and the dimension of feature representation is reduced significantly. PCA can also remove the noise existed in the input data. Moreover, Liu et al. [17, 18] propose multi-task method for multiple/single view human action grouping and recognition. Xu et al. [19] introduce a Multi-modal and Multi-view Interactive dataset to solve the problem of cross domain. Liu et al. [20] propose a human action recognition method via coupled hidden conditional random fields model. Yang et al. [21] and Zhu et al. [22] propose some methods to improve the quality of images and videos. Gao et al. [23, 24] and Ji et al. [25] introduce some learning method for 3D Object Retrieval and classification. Lu et al. [26, 27] propose some hash and semi-supervised learning methods.

Deep learning methods lead to another branch of the learning method. They usually use convolutional neural networks with multiple layers to learn the feature representation. Recently, deep learning technology has achieved huge success in image recognition and analysis [28], which attracts a lot of efforts from both academia and industry. As its extension,

video recognition research has been largely driven by the advances in image recognition methods [29].

Karpathy et al. [15] use convolutional neural networks (CNNs) to process the task of large-scale video classification. They provide extensive experimental evaluations of multiple approaches in order to extend CNNs into the video classification. The algorithm processes the input at two spatial resolutions, improving the runtime performance of CNNs at no cost in accuracy. Such a method obtains significant performance improvement when applying their networks to the UCF-101 dataset. Simonyan et al. [29] exploit a Two-Stream ConvNets model to incorporate spatial and temporal networks simultaneously, in which video frames and optical flow are put into the two streams respectively. Finally, the integration of spatial and temporal information improves the accuracy of video classification. Ryoo et al. [30] introduce a Pooled times series representation called PoT, which captures ego motion information in first-person videos. The idea is to keep track of the change of element in per-frame descriptor vectors over time. The algorithm applies multiple pooling operators to the time series, and generates the efficient PoT feature representation for a video. Wang et al. [31] come up with a novel video feature descriptors called TDD, which combines the deep learning method and temporal handcraft features. They first set up Two-Stream ConvNets to extract the deep learned feature maps, then extract the TDD feature descriptors by using feature map normalization and trajectory pooling operation. Specifically, in the trajectory pooling step, the deep learned feature map and handcrafted local features are linked closely by the trajectory pooling operation.

Regarding action recognition task, an interesting but important question is how to define an action. The definitions of event and action are unfortunately ambiguous in most computer vision literature [32]. As a result, there is a "*gap*" between the action described by people and the essential features in video, which is an obstacle to learn the valid feature representation.

3 Attribute-Based Supervised Deep Learning Model

In this section, we discuss the main contributions of the paper after explaining the motivation of our scheme. Then, we demonstrate the limitations of current ConvNets from the perspective of a practical action recognition task. We show the system overview with respect to our Attribute-Based Supervised Deep Learning Model, and finally describe our idea in a formulating way.

3.1 Motivation

This idea originates from the essence of action recognition task. Action classification is much more complex than image classification due to the dynamics of objects, scenes and trajectories, which all confuse people when determining the action category.

In the traditional ConvNets, the models usually extract the appearance features from the input data. However, some videos with similar appearances may belong to totally different categories, which may result in a wrong classification. To solve this problem, adding some high-level semantic information to ConvNets will be helpful. It is better for ConvNets to process their learning procedures under the guidance of the semantic information. That is why we introduce the attributes information.

3.2 Challenges

Our challenge comes from a true dilemma that we encountered in the action recognition task: when recognizing the actions from UCF101 dataset [33], we notice that most video categories with high classification accuracy have relatively simple background, however, the classification accuracy is generally low if the video's background is complex and varying, as shown in Fig. 2.

Fig. 2 Different background conditions in video. (a)–(e) show a Surfing action with similar backgrounds, and (f)–(i) demonstrate the action of Pizza Tossing under various background conditions

To intuitively interpret this problem, we conduct several experiments, as depicted in Fig. 3, where we directly feed the raw images covering various backgrounds into the CNNs. The observations can be summarized as follows:

- If the previous several input frames contain bright background, it is easy to set the parameters for ConvNets.
- If there are some images in the sequence belonging to the same category with dark background, the ConvNets have to adjust their parameters to fit in the new circumstance.

- However, if there are frames with interlaced bright background and dark background, which means the background varies with relatively large difference, the ConvNets will get confused and eventually end up with a poor performance. Compared to the case where the background is slightly changed, the last case illustrates that the ConvNets cannot handle the situation where large changes occur in the background. For this case, it is difficult for the ConvNets to adjust their parameters correctly.

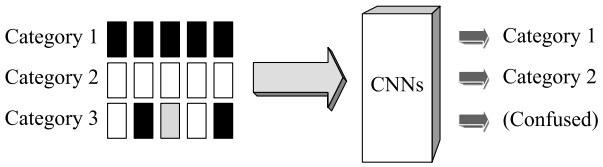

Fig. 3 The simple exemplars when CNNs get confused by the inputs (Here, each rectangle represents one video. In the first two categories, the backgrounds have only minor changes in one action category while the background changes dramatically in Category 3. The observation is that CNNs get confused and do not adjust the parameters properly in the last case)

The key factor of this problem is that some features are not able to truly distinguish the current action category and the other categories. The pixel-level convolutional operation used in ConvNets mainly investigates the spatial relation of the pixels, which may not be suitable for extracting temporal features.

In order to enhance feature representation, we propose to convey some extra message to ConvNets. An intuitive idea is using the attribute to describe the input data, and to guide the learning procedure of ConvNets.

3.3 System overview

Based on the analysis above, we propose our Attribute-Based Supervised Deep Leaning model, which is illustrated in Fig. 4.

Compared to the original ConvNets model employed in most existing works, our model has some complementary in both input and output modules of the network. We add some attributes attached to the category of the videos, which means that the videos in the same category share the same value in the attribute space. When processing the input data, we put the input data and their labeled category into network as usual, on top of it, we also add the attribute values for each input data. At the output layer, we extend the traditional category output layer to several parallel attribute output layers, which represent the feature map in corresponding attribute field respectively. The components in the green box represent the standard skeleton for ConvNets whereas the components in the red box show how we add the attribute information.

3.4 Formulation of the proposed idea

In the Attribute-Based Supervised Deep Learning Model, each training sample is denoted as

$$data_t = \{I, P\}, \tag{1}$$

where I is the input data and P is the prediction target. Specifically, we can utilize a single frame or frame difference as the input I.

In this paper, we propose to construct an extensive prediction target, which is

$$P = \{C, A_1, A_2, \ldots\}. \tag{2}$$

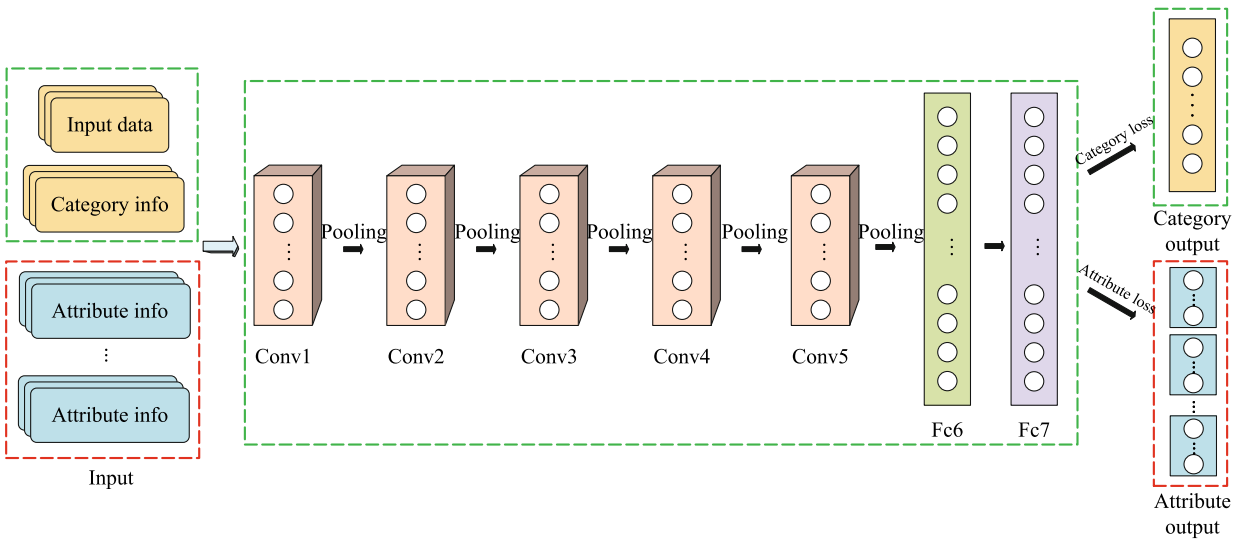

Fig. 4 The basic structure of attribute-based supervised ConvNets

Here, $C \in \{0,1\}^c$ is the category vector indicating the membership between the training sample and c categories, and $A_i \in \{0,1\}^{(a_i)}$ denotes the vector for the ith kind of attribute indicating the relationship between the sample and a_i attributes. We will introduce how to construct A_i in more details later in the next section.

Overall, the basic loss function is defined as:

$$\overline{\zeta} = -\frac{1}{m}\left\{\sum_{i=1}^{m}\left[\sum_{æ=1}^{k} y^{(iæ)} * \log(h_æ(x^{(i)}))\right]\right\}, \quad (3)$$

where m is the number of data, k is the category range, $y^{(iæ)}$ means the ground truth of data i, and $h_æ(x^i)$ denotes the output in j column of data i.

Our loss function is:

$$\overline{\zeta_T} = \overline{\zeta_C} + \sum_{p}^{r} \varsigma_p \overline{\zeta_{A^p}}, \quad (4)$$

where r is the number of attributes, $\overline{\zeta_T}$ is the total loss value, $\overline{\zeta_C}$ is the origin loss for category classification and $\overline{\zeta_{A^p}}$ is the loss for attribute p.

In the conventional deep learning model, only the category vector is utilized in the training phase, i.e., we have $P = \{C\}$ and $\overline{\zeta_T} = \overline{\zeta_C}$. As we have discussed above, incorporating the attributes as the side information can improve the performance, which definitely differs to the existing works.

4 Implement details

In this section, we introduce our implementation details about the Attribute-Based Supervised Deep Learning Model, including the network structure, preparation for input data and other important issues.

First of all, we implement the basic ConvNets model and other two extra derived ConvNets models. Based on it, we further construct attribute output layers in ConvNets to supervise the ConvNets using backward propagation. Before the training stage starts, videos are notated with some tags. In our implementation, we only define attributes at the category level rather than at the video level, which significantly reduces the workload. Then, we transport these data with their attributes into the Attribute-Based Supervised Deep Learning Model. At the testing stage, attribute information is invisible, and we obtain the category output layer as a foundation for classification results.

4.1 Basic ConvNets structure

The structure of our model mainly refers to the basic ConvNet used in ImageNet [34] classification. The ConvNets exploit single frames as the input, where a 224×224 sub-image is randomly cropped from the selected frame. There are five convolutional layers followed by two full connect layers, and the last category output layers contain the classification information. We set the initial learning rate to 10^{-3} and decay 5×10^{-4} after each $10k$ iteration. To make ConvNets converge quickly, we train our ConvNets on the pre-trained Caffe [35] model of ImageNet [34].

4.2 Input data preparation

With respect to the spatial data information, we use the single frame image serials sample for each video and choose $L = 10$, which is the number of single frames, to extract our input images from each video. Regarding the temporal information, we exploit the frame difference between consecutive frames due to its low-computational cost and obvious temporal characteristic. We abandon the dense trajectory or optical flow features [36], because such features are not easy to extract in terms of the computational load. Additionally, we aim at investigating the effect of adding attribute information in the CNNs framework, rather than obtaining a high classification accuracy, so we choose the simple and intuitive frame difference. The simpler our input data is, the easier for us to analyze the effectiveness caused by the attributes.

The serials of frame differences are calculated as follows. In a video, we extract the images and their subsequent frame images. At each pixel position, we calculate the color RGB value difference in each channel, respectively. Here, we adopt the tolerance and amplification mechanism to remove noise in the image, and stress out the variety area. For the last image in the video, we consider the first image as its next image.

We use the single frame (SF) and frame difference (FD) as the input data for a basic ConvNets alone, and use them together in the Two-Stream ConvNets structure as shown in Fig. 5. We expect it to reflect the impact in different domain information caused by attribute supervision.

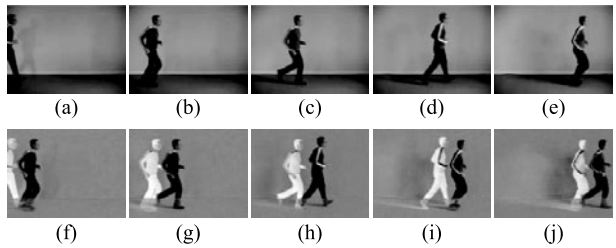

Fig. 5 (a)–(e) Single frame and (f)–(j) frame difference

4.3 Extra network structure implementation

4.3.1 Group Pooling Model

The Group Pooling ConvNets model is setup according to the

the model proposed in Ref. [37], which optimizes the ConvNets by max pooling operation. We arrange the input data group by group, which means that continuous L ($L = 10$) frames come from the same video. Afterwards, we add a custom Group-Pooling layer behind the 5th pooling layer to aggregate all feature maps in one video. The operation of group pooling would take out the max value at each position of the feature map for all images in a video. Finally, the max-pooling results are propagated forward into the 6th full-connected layer. Other similar structures in the ConvNets can also be adopted.

4.3.2 Two-Stream Model

This Two-Stream ConvNets model is implemented according to the model in Two-Stream ConvNets [29]. Basic spatial networks are constructed as suggested in Section 4.1, which is used to capture the image static appearance cues. Structures of another temporal stream are similar to the spatial stream, but they take in the frame difference as the input data. Finally, we manage to enable the two streams converge after the two fc7 layers using a concatenation layer.

4.4 Classification policy

Since the direct output of ConvNets is the frame feature map and a video may consist of many of these features, there are many methods that can be chose for the final classification. One of the intuitive approaches is to classify each image into its category then to adopt a voting mechanism to pick up the category for a video. An alternative is to take out the value at the last layer of ConvNets as the feature representation, then stack all the features for a video into a long global feature vector, and eventually train a linear SVM model as the classifier.

It is noted that the output layers of Group Pooling ConvNets model are identical for all images in the same video due to the pooling operation. If we put the feature representation into a SVM, its length is $1/10$ of others, thus leading to a pretty bad result. Therefore, we do not perform the SVM policy for Group Pooling Model.

4.5 Attribute-Based Supervised Deep Learning Model

On the basis of above foundation work, we establish several parallel attributes output layers with the last category output layer, and connect them with each corresponding ground truth label layer as a loss layer. In this way, the input label information can control the attribute output layers by the loss function, and both the category output layer and the attribute output layers are able to regulate the fc7 layer in ConvNets by backward propagation. That is how the supervision function is realized.

To help on understanding the concept, we give a concrete example. For UCF101 dataset [33], we use two types of attributes: single background and upper classes. The former one is the sub-category information while the latter one is the super-category information. Each video has two possible values (either 0 or 1 on single background attribute), which indicates that whether the background varies in the category or not. Similarly, each type of video belongs to one upper class of five possible values described in UCF101 dataset, which are Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments and Sports. We add two attribute output layers with two and five output nodes respectively, in which each of them is connected to a label layer by a loss layer. The input data can be described as $(data, C, A_1, A_2)$, where *data* indicates the path to image file, C refers to the category from 101 categories, A_1 is the 0 or 1 about whether background is single, and A_2 is the ID of upper class for this video.

5 Experiments

We evaluate our proposed model on the popular video action datasets KTH [38], UCF101 [33] and HMDB51 [39], according to the standard procedure. In this section, we present our experiment plans and results, then discuss some observations.

5.1 Dataset

5.1.1 UCF101

UCF101 [33] is a common action recognition dataset collecting realistic action videos. There are totally 13 320 videos of 101 action categories in this dataset. It has large diversity and variation in camera motion, object appearance, human pose and object scale, which provides the possibility for the recognition algorithms to test and verify their robustness and effectiveness in the realistic situation. We split the dataset into training and testing data according to the standard rules.

5.1.2 KTH

KTH [38] is a video database containing six types of human actions (walking, jogging, running, hand waving and hand clapping). The tiny differences between action categories specially require the precise feature extraction ability of algorithm.

5.1.3 HMDB51

HMDB51 [39] is provided by Serre Lab. The videos are collected from various sources, mostly from movies. Compared to UCF101 dataset, the action categories in HMDB51 are much more highly summarized. There can be various different scenes and activities rolled in these videos, and the only common characteristic is that the protagonist has made the target action in the video.

Most of current methods could hardly achieve over 50% classification accuracy in HMDB51, reflecting its challenge for most of action recognition approaches. Following the standard procedure, we split the dataset into training and testing sets.

5.2 Baseline

Our baseline algorithm is just the basic ConvNets proposed by Karpathy et al. [15], as we intend to show how the performance can be improved by adding attributes into the network. For some datasets, such as UCF101, we also compare our algorithm with existing algorithms including Two-Stream [29] and TDD [31].

For KTH [38] and HMDB51 dataset [39], although there are many relevant research results, most of them are based on hand-crafted feature descriptors or complex ConvNets models. It is unsuitable to compare the classification accuracy under different conditions. We have been taught in the primary school that the most important principle in experiments is to control the variables. In order to observe the effectiveness of the attribute supervision, we use the results of the same basic ConvNets structure in UCF101 dataset [33] as the baseline, and then we add extra attribute layers and inspect the positive effect caused by attribute supervision.

5.3 Experiment details

We use an open-source deep learning framework Caffe [35] to implement our convolutional neural network model. The basic input data for our Attribute-Based Supervised Deep Learning Model is serials of single frame and frame difference, which have been introduced in Section 4.2. For the three datasets mentioned above, we adopt the similar policy to deal with the videos, that is, splitting the training and testing set, extracting static frames from video and calculating the frame difference for each frame.

With Caffe [35] tools, we establish the basic ConvNets similar to the model used for ImageNet [34] classification task, which is convenient for us to execute the fine tune operation later. Besides the basic network structure, we modify the source code in Caffe [35] to achieve two extra networks. The first one is the Group Pooling Model, executing max pooling operation after the 5th pooling layer among all the frames within one video. The other one is the Two-Stream Model, referring to the idea of Two-Stream ConvNets [29]. We use a contacting layer to put the spatial and temporal feature map together. We have tried many places to add the concatenation layer, finally chosen the position after fc7 layer.

When obtaining our model, we have two policies to use the ConvNets output. One is to use the softmax layer directly followed by a vote strategy among video frames to calculate the final result. Another solution is taking out the softmax layer value and stacking them to a long vector as the feature representation for a video. We also achieve a plan which stacks the single frame and frame difference to one vector as the SVM input. The tool we use to train the SVM model is called LibSVM [40]. Finally, we use the SVM model as a classifier to execute action recognition task.

After the above preparation step for baseline and benchmark, finally we add the attribute layers to implement our Attribute-based Supervise Deep Learning Model. As described in Section 4, we use several attribute output layers to represent certain attribute values, and use corresponding label layers as the ground truth to supervise the learning procedure.

The types of attributes for each dataset are introduced as follows. For UCF101 dataset [33], we find that some types of videos have similar background within category while others do not, and those classification results for the videos with various backgrounds are always not good. In view of this, we choose an attribute to indicate whether background is single within a category. Besides, we get five super class information from the dataset official information, we also use them as another type of attribute.

For KTH [38], we select an attribute that indicates whether the persons in video are moving. In the first three types of videos, people only stand on the spot doing some action, while the other three are just the opposite.

For HMDB51 [39], almost every category of video contains various background. That is possibly why most deep learning methods have unsatisfactory effect on HMDB51. In this case, denoting attribute to the background is meaningless. Fortunately, we find the the super class information in the official website of HMDB51, and we thus use them as the only attribute for HMDB51.

Table 1 shows the overview about the experimental results.

Table 1 A overview about experimental results

Dataset	UCF101/%	KTH/%	HMDB51/%
Baseline	63.30	-	-
BC	64.74	79.00	34.38
BC+Attr	66.64	81.23	36.14
BC+SVM	69.71	**88.76**	37.56
BC+Attr+SVM	**71.48**	87.95	**38.69**

In Table 1, "BC" represents the "Basic CNNs" and "Attr" represents "Attribute". "BC+Attr" means that we use Basic CNNs with attribute supervision and "BC+SVM" means that we use the feature maps from Basic CNNs to train a SVM model. "BC+Attr+SVM" means we combine the policy of Basic CNNs with attribute supervision and SVM model.

Table 2 shows the comparison of our method with other popular methods.

Table 2 Comparison with other methods in UCF101

Type	Method	Accuracy/%
None trajectories	STIP+BoVW [33] (2012)	43.6
	Karpathy et al. [15] (2014)	63.3
	MDI [41] (2016)	70.9
	Our Method	**71.48**
	Ensemble-based(RSM) [42] (2015)	75.05
Trajectories-based	Two-Stream [29] (2014)	88.0
	TDD [31] (2015)	91.5

Our method can obtain a minor improvement, compared to basic deep feature methods. However, our performance is lower than the method, called Ensemble-based (RSM), which uses the random subspace method (RSM) [43] to divide all features into random group and then integrates multiple classifiers in the framework.

Apparently, trajectories-based methods, like optical flow, iDT and Trajectory-Pooled descriptors, obtain much better results due to the fact that trajectory can be considered as a sort of semantic level feature. However, extracting such features requires additional computation load. In our system, we only pick up simple attributes from video in spatial domain, which is not comparable to the complicated features.

5.4 Exploring attribute effect on different aspects

5.4.1 Attribute effect on spatial/temporal feature

Firstly, we evaluate the effect of attribute on the basic network, where input data are chosen to be spatial frames and temporal frame differences. The experimental results can be seen in Table 3, where SF represents Single Frame, FD refers to Frame Difference, attr1 indicates whether background has less change and attr2 is the super class attribute.

Table 3 Result/% of spatial/temporal basic ConvNets for UCF101

Scheme	BC	BC+attr1	BC+attr2	BC+attr1&2
SF	64.74	65.08	66.35	**66.64**
FD	57.74	64.05	63.39	**65.35**
SF + SVM	64.97	-	-	**66.83**
FD + SVM	**64.90**	-	-	64.53
SF+FD+SVM	69.71	-	-	**71.48**

In Table 3, "attr1" and "attr2" represent the attribute "single background" and "upper class" respectively. "SF" and "FD" indicate the "Single Frame" and "Frame Difference". "SF+SVM" means we use Single Frame as input data and train SVM model with the feature maps, "FD+SVM" and "SF+FD" are similar.

Seen from the results, using SVM leads to better result than conducting classification in the last output layer directly. Meanwhile, attribute layers have positive help for deep learning procedure. The attributes in relation to background and super class both contribute to the performance, and their combination achieves even better result. We believe that there still exist many semantic attributes that we haven't adopted in this experiment.

The result of dataset KTH and HMDB51 can be viewed in Tables 4 and Table 5. The meanings of abbreviations are as same as those in Table 3.

Table 4 Result/% of spatial /temporal basic ConvNets for KTH

Dataset	BC	BC+Attr
SF	79.00	**81.23**
FD	85.00	**87.49**
SF + SVM	84.13	**86.33**
FD + SVM	**89.46**	87.60
SF+FD+SVM	87.76	**87.95**

Table 5 Result/% of spatial/temporal basic ConvNets for HMDB51

Dataset	BC	BC+Attr
SF	34.38	**36.14**
FD	**32.55**	31.90
SF + SVM	35.10	**36.41**
FD + SVM	**33.46**	32.75
SF+FD+SVM	38.56	**38.69**

5.4.2 Attribute effect on pooling feature

We conduct our attribute-based supervised model for Group Pooling Model. The results are listed in Table 6.

In Table 6, "BC" is for "Basic CNNs", "PM" represents "Pooling Model", and "PM+Attr" represents the Pooling Model with attribute supervision.

We notice that the effectiveness of using attribute is not clear for the Pooling Model. In KTH and HMDB51 dataset,

we obtain even worse results when adding attributes. Another interesting observation is that when we add the background attribute to UCF101 dataset on max-pooling feature, it fails to train a model, which means that the network never converges. Therefore, we only use the super-class attribute. From the result above, it implies that the max pooling operation may change the video frames into incredible feature maps, which are invisible and cannot be explained by semantic description. Consequently, it is no longer suitable to be described by our handcrafted attributes.

Table 6 Result of Pooling Model for datasets

Dataset	UCF101		KTH		hMDB51	
Scheme	SF/%	DF/%	SF/%	DF/%	SF/%	DF/%
BC	64.74	57.74	79.00	85.00	34.38	32.55
PM	67.01	66.22	**85.16**	**88.00**	**39.61**	35.10
PM+Attr	**68.99**	**66.35**	82.16	85.28	37.12	**35.21**

5.4.3 Attribute effect on Two-Stream feature

We also apply our plan to Two-Stream Model, and the results are in Table 7.

Table 7 Result of Two-Stream ConvNets for datasets

Dataset	UCF101/%	KTH/%	HMDB51/%
BC	64.74	79.00	34.38
TS	63.10	**79.23**	32.35
TS+Attr	**65.13**	77.63	**34.48**

In Table 7, "BC" is also for "Basic CNNs", "TS" represents "Two Stream" and "TS+Attr" represents the Two Stream Model with attribute supervision.

From the above table we can find out that attribute-based supervision has positive effect on this Two-Stream network, except for the KTH dataset. One possible reason may be that our temporal input data are not very useful. Nevertheless, as we explained previously, attributes play a supervision role here to govern the ConvNets how to learn better. It has to know which features should be learned or not. Two-Stream model provides with more information than single network. Therefore, in Two-Stream network, ConvNets have more choices at the same time, and they can easily pick up the valid feature maps from a temporal stream or a spatial stream or a part of them, according to the guidance by attributes. That is the mission we expect the attribute to accomplish.

5.5 Analysis

As the results above show, our Attribute-Based Supervised Deep Learning Model outperforms some basic deep learning models, such as Deep Nets [15]. That means our way of using attribute has indeed helped the deep learning procedure.

Our experiments demonstrate that among deep learning methods, utilizing attribute to supervise the learning procedure can promote the ability of Deep ConvNets further. We feel that our performance improvement is impeded because of two restrictions that existed in the current scheme. Firstly, our ConvNets structure is only based on an ordinary network, and we have not designed each layer with careful adjustment for the certain dataset. Additionally, our selected attributes are rather simple. We only select a small number of category level labels to avoid lots of tagging work. If we add more valuable attributes which depict the action precisely and explore their best combinations, the experimental results can be much better.

6 Conclusion

Deep learning model is popular in the field of computer vision, and it is a practical solution for action recognition task. In this paper, we proposed a novel semi-supervised deep learning model, called Attribute-Based Supervised Deep Learning Model, which redeems the drawbacks of basic ConvNets for recognizing the realistic actions. Attribute-Based Supervised Deep Learning Model takes attributes of input data as a sort of known information to supervise and guide the learning process of a convolutional neural network. We evaluated our proposed models on three challenging datasets, and compared the performance with basic ConvNets methods. Experiments show that with only a small change to receive attribute information, ConvNets can promote the classification execution accuracy in various situations. Therefore, it is convincing that, with attribute-based supervised, ConvNets can extract more precise features and improve the performance of that action recognition.

References

1. Lao W L, Han J G. Automatic video-based human motion analyzer for consumer surveillance system. IEEE Transactions on Consumer Electronics, 2009, 55(2): 591–598
2. Zhang B C, Alessandro P, Li Z G, Vittorio M, Liu J Z, Ji R R. Bounding multiple gaussians uncertainty with application to object tracking. International Journal of Computer Vision, 2016, 1–16
3. Chen C, Liu M Y, Zhang B C, Han J G, Jiang J J, Liu H. 3D action recognition using multi-temporal depth motion maps and fisher vector. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence. 2016, 3331–3337
4. Han J G, Dirk F, De With P H N. Broadcast court-net sports video analysis using fast 3-D camera modeling. IEEE Transactions on Cir-

cuits and Systems for Video Technology, 2008, 18(11): 1628–1638

5. Ding G G, Guo Y C, Zhou J L, Gao Y. Large-scale cross-modality search via collective matrix factorization hashing. IEEE Transactions on Image Processing, 2016, 25(11): 5427–5440
6. Lin Z J, Ding G G, Han J G, Wang J M. Cross-view retrieval via probability-based semantics-preserving hashing. IEEE Transactions on Cybernetics, 2016
7. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005, 886–893
8. Laptev I, Marszałek M, Schmid C, Rozenfeld B. Learning realistic human actions from movies. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2008, 1–8
9. Dalal N, Triggs B, Schmid C. Human detection using oriented histograms of flow and appearance. In: Proceedings of European Conference on Computer Vision. 2006, 428–441
10. Wang H, Schmid C. Action recognition with improved trajectories. In: Proceedings of IEEE International Conference on Computer Vision. 2013, 3551–3558
11. Li F F, Pietro P. A bayesian hierarchical model for learning natural scene categories. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005, 524–531
12. Lee H, Battle A, Raina R, Ng A Y. Efficient sparse coding algorithms. In: Proceedings of Advances in Neural Information Processing Systems. 2006, 801–808
13. Yang Y, Wang X, Liu Q, Xu M L, Yu L. A bundled-optimization model of multiview dense depth map synthesis for dynamic scene reconstruction. Information Sciences, 2015, 320: 306–319
14. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems. 2012, 1097–1105
15. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Li F F. Large-scale video classification with convolutional neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2014, 1725–1732
16. Price A L, Patterson N J, Plenge R M, Weinblatt M E, Shadick N A, Reich D. Principal components analysis corrects for stratification in genome-wide association studies. Nature Genetics, 2006, 38(8): 904–909
17. Liu A A, Su Y T, Jia P P, Gao Z, Hao T, Yang Z X. Multipe/single-view human action recognition via part-induced multitask structural learning. IEEE Transactions on Cybernetics, 2015, 45(6): 1194–1208
18. Liu A A, Xu N, Su Y T, Lin H, Hao T, Yang Z X. Single/multi-view human action recognition via regularized multi-task learning. Neurocomputing, 2015, 151: 544–553
19. Xu N, Liu A A, Nie W Z, Wong Y Y, Li F W, Su Y T. Multi-modal & multi-view & interactive benchmark dataset for human action recognition. In: Proceedings of the 23rd ACM International Conference on Multimedia. 2015, 1195–1198
20. Liu A A, Nie W Z, Su Y T, Ma L, Hao T, Yang Z X. Coupled hidden conditional random fields for RGB-D human action recognition. Signal Processing, 2015, 112: 74–82
21. Yang Y, Wang X, Guan T, Shen J L, Yu L. A multi-dimensional image quality prediction model for user-generated images in social networks. Information Sciences, 2014, 281: 601–610
22. Zhu Y M, Li K, Jiang J M. Video super-resolution based on automatic key-frame selection and feature-guided variational optical flow. Signal Processing: Image Communication, 2014, 29(8): 875–886
23. Gao Y, Wang M, Tao D C, Ji R R, Dai Q H. 3-D object retrieval and recognition with hypergraph analysis. IEEE Transactions on Image Processing, 2012, 21(9): 4290–4303
24. Gao Y, Wang M, Ji R R, Wu X D, Dai Q H. 3-D object retrieval with hausdorff distance learning. IEEE Transactions on Industrial Electronics, 2014, 61(4): 2088–2098
25. Ji R R, Gao Y, Hong R C, Liu Q, Tao D C, Li X L. Spectral-spatial constraint hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 2014, 52(3): 1811–1824
26. Lu X Q, Zheng X T, Li X L. Latent semantic minimal hashing for image retrieval. IEEE Transactions on Image Processing, 2016, 26(1): 355–368
27. Lu X Q, Li X L, Mou L C. Semi-supervised multitask learning for scene recognition. IEEE Transactions on Cybernetics, 2015, 45(9): 1967–1976
28. Zhang D W, Han J W, Han J G, Shao L. Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(6): 1163–1176
29. Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. In: Proceedings of Advances in Neural Information Processing Systems. 2014, 568–576
30. Ryoo M S, Rothrock B, Matthies L. Pooled motion features for first-person videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, 896–904
31. Wang L M, Qiao Y, Tang X O. Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, 4305–4314
32. Liu J G, Yu Q, Javed O, Ali S, Tamrakar A, Divakaran A, Cheng H, Sawhney H. Video event recognition using concept attributes. In: Proceedings of IEEE Workshop on Applications of Computer Vision. 2013, 339–346
33. Soomro K, Zamir A R, Shah M. Ucf101: a dataset of 101 human actions classes from videos in the wild. 2012, arXiv preprint arXiv:1212.0402
34. Deng J, Dong W, Socher R, Li L J, Li K, Li F F. Imagenet: A large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2009, 248–255
35. Jia Y Q, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia. 2014, 675–678
36. Wang H, Kläser A, Schmid C, Liu C L. Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision, 2013, 103(1): 60–79
37. Ng J Y H, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G. Beyond short snippets: deep networks for video classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2015, 4694–4702
38. Schuldt C, Laptev I, Caputo B. Recognizing human actions: a local svm approach. In: Proceedings of the 17th International Conference

on Pattern Recognition. 2004, 32–36

39. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. Hmdb: a large video database for human motion recognition. In: Proceedings of IEEE International Conference on Computer Vision. 2011, 2556–2563
40. Chang C C, Lin C J. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): 27
41. Bilen H, Fernando B, Gavves E, Vedaldi A, Gould S. Dynamic image networks for action recognition. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition. 2016
42. Bagheri M, Gao Q G, Escalera S, Clapes A, Nasrollahi K, Holte M, Moeslund T. Keep it accurate and diverse: enhancing action recognition performance by ensemble learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015, 22–29
43. Ho T K. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 832–844

Kai Chen received the BS degree from the School of Software, Tsinghua University, China in 2014, where he is currently pursuing the MS degree with the School of Software. His research interests include multimedia information retrieval, computer vision, and machine learning.

Guiguang Ding received the PhD degree in electronic engineering from Xidian University, China. He is currently an associate professor with the School of Software, Tsinghua University, China. His current research focuses on the area of multimedia information retrieval and management, in particular, visual object classification, automatic semantic annotation, content-based multimedia indexing, social multimedia retrieval, mining and recommendation. He has published about 40 research papers in international conferences and journals and applied for eight Patent Rights in China.

Jungong Han is a senior lecturer with the Department of Computer Science at Northumbria University, UK. Previously, he was a senior scientist (2012–2015) with Civolution Technology (a combining synergy of Philips CI and Thomson STS), a research staff (2010–2012) with the Centre for Mathematics and Computer Science, and a researcher (2005–2010) with the Technical University of Eindhoven in Netherlands. Dr. Han's research interests include multimedia content identification, computer vision, and artificial intelligence. He has written and co-authored over 100 papers, in which one first-authored paper has been cited, up to date, for more than 500 times. He is an associate editor of Elsevier Neurocomputing and Springer Multimedia Tools and Applications.