Binary Hashing CNN Features for Action Recognition

Weisheng Li[1]*, Chen Feng[1], Bin Xiao[2], Yanquan Chen[2]

1 Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[e-mail: liws@cqupt.edu.cn]

2 college of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[e-mail: xiaobin@cqupt.edu.cn]

*Corresponding author: Weisheng Li

Received December 3, 2017; revised March 4, 2018; revised April 3, 2018; accepted April 23, 2018; published September 30, 2018

Abstract

The purpose of this work is to solve the problem of representing an entire video using Convolutional Neural Network (CNN) features for human action recognition. Recently, due to insufficient GPU memory, it has been difficult to take the whole video as the input of the CNN for end-to-end learning. A typical method is to use sampled video frames as inputs and corresponding labels as supervision. One major issue of this popular approach is that the local samples may not contain the information indicated by the global labels and sufficient motion information. To address this issue, we propose a binary hashing method to enhance the local feature extractors. First, we extract the local features and aggregate them into global features using maximum/minimum pooling. Second, we use the binary hashing method to capture the motion features. Finally, we concatenate the hashing features with global features using different normalization methods to train the classifier. Experimental results on the JHMDB and MPII-Cooking datasets show that, for these new local features, binary hashing mapping on the sparsely sampled features led to significant performance improvements.

Keywords: Action Recognition, CNN Feature, Binary Hashing, Feature Normalization

1. Introduction

Human action recognition aims to automatically classify the ongoing action in a video clip, which is one of the most important areas in video analysis and computer vision research. Its potential applications include video surveillance, human computer interaction, content-based retrieval, and so on [1]. It is one of the challenging problems in computer vision for some reasons. First, the viewpoint changes, background clutter and motion characteristics contain large intra-class variations, even within one action class. Second, the identification of an action class is related to many other high-level visual clues, such as human poses, the scene class and interacting objects. These related problems are very difficult to solve. Furthermore, although videos are temporally segmented, the segmentation of an action is more subjective than the segmentation of a static object, which means that there is no precise definition of the beginning and end of an action. Finally, the high dimension and low quality of video data usually adds difficulty to develop robust and efficient recognition methods.

Action recognition in videos has attracted considerable attention of researchers in the past few years, and much progress has been made in computer vision field [2-8]. Basically, a number of existing methods [9-11] perform the steps for action recognition, including feature extraction, feature encoding and classifier training. Feature encoding is arguably the most important since each action can only be discriminatively represented by proper feature encoding. This is crucial to guarantee intra-class and inter-class separability during the recognition step itself.

With the resurgence of efficient deep learning and pose estimation algorithms, several works have focused on how to combine local features with high-level information (e.g., pose information) and learned features. These include the two-stream convolutional networks based on RGB appearance frames and optical flows [12], the two-stream convolutional networks combined with dense trajectories [8], Pose Convolutional Neural Networks (P-CNN) [13], recurrent pose-attention network (RPAN)[14], and skeleton sequence-based multi-stream networks[15].

Generally, deep architectures for action recognition in videos take as their input short video clips consisting of a series of frames. The use of single frames might be insufficient to effectively capture the dynamics of actions since single frames ignore temporal information. However, using longer video clips requires more model parameters and demands more training data and computational resources. This problem also exists in other popular CNN architectures, such as 3D CNNs [16]. Thus, state-of-the-art deep action recognition models are usually trained to generate useful features from short video clips. These models are then pooled to generate holistic sequence-level descriptors, which are then used to train a linear classifier with specific action labels.

For example, in the P-CNN model, a video representation is obtained by extracting the output features of the FC (fully connected) layers from the RGB and optical flow streams,

and these are combined using max/min pooling. Note that the max/min pooling captures only the first-order correlations between the features. A higher-order pooling [17] that captures higher order correlations between the CNN features can be more appropriate, which is the main motivation for the scheme proposed in this paper.

Fig. 1. Two different actions : cutting a tomato (left) and cutting a carrot(right)

Generally, action recognition is a coarse-grained problem with the goal of distinguishing between human actions under different scene conditions (e.g. running on a track vs. swimming in water). However, in this work, we consider fine-grained action recognition, such as the act of cutting a tomato vs. cutting a carrot illustrated in **Fig. 1**. As can be seen, the coarse-grained action of cutting is still similar. However, the underlying finer-grained concept of the object being cut is totally different. Understandably, detecting such actions poses greater challenges and thus, requires a different treatment. Specifically, we assume a two-stream CNN framework with human pose estimation as suggested in [13] with separated RGB appearances and optical flow streams. We then extract CNN features from different parts of the human body in each stream and proceed to concatenate the final video features using different aggregation methods.

As mentioned above, while CNN features at the frame-level might be very noisy, we assume that the correlations among the temporal evolutions of frame features can capture useful action cues that may help improve recognition performance. Intuitively, some of the actions might have key sub-action frames for better discrimination of the action, while they may be ignored as noise when the max/min pooling scheme is used. In this paper, we use this intuition to develop a theoretical framework for action recognition using hashing pooling on the two-stream CNN features. Our pooling scheme is based on binary hashing. It is a simple technique that decomposes a video frame’s feature matrix, which is computed from the FC layer input. After inputting a sequence of video frames and their corresponding joint

positions [18], we can extract the CNN features from each frame in each part of the human body. After the CNN feature extraction is prepared, we select some key frames with their features from the video and compare them to the adjacent feature vector in the chosen frames. Then, we use a vector with a binary value to represent the results of the feature comparison between one pair of adjacent frames. Since we selected some frames from a video, we can obtain a binary value matrix after comparisons. Afterwards, we can compose the binary value matrix into a decimal vector and get the raw hashing feature of the video. Obviously, the result in the decimal vector has a wide range compared to the P-CNN features, and we must normalize the hashing features if we want to fuse these two features. Following [3], we tried L_2 normalization. We considered that the normalization approach influences the final classification result, and we also tried L_1 normalization and the fusion of $L_1+\beta \cdot L_2$ normalization.

The rest of this paper is organized as follows. Section 2 introduces the related works on fine-grained action recognition and the video representation approach. Section 3 describes the details of our B-CNN method. We present the experiments in Section 4 by using the proposed method on two standard action datasets. The conclusion is presented in Section 5.

2. Related Work

Existing approaches to fine-grained action recognition have been direct extensions of methods developed to address image recognition problems and have been mainly based on handcrafted features. A few notable approaches [9-11] first extract features from spatial-temporal interest point locations and then train a classifier using feature fusion. However, more recent works [12-16] have advanced towards data-driven deep feature learning approaches.

As mentioned above, the lack of sufficient annotated video data, and the need for expensive computational devices make direct extension of these methods (which were primarily developed for image recognition tasks) challenging for video data, thus demanding efficient representations.

Another promising setting for fine-grained action recognition uses mid-level features, such as human poses. Obviously, estimating human poses and developing action recognition frameworks based on them directly tackles the action inference from processing on the pixel-level, thereby allowing for a higher-level of sophisticated action reasoning. Although there have been significant advancements in pose estimation recently, most of these methods are computationally demanding and thus difficult to extend to the millions of video frames that form standard datasets.

A different approach to fine-grained recognition was proposed by Zhou et al [19] to detect and analyze human-object interactions in videos. Their method begins by generating regional proposals for human object interactions in the scene, extracts the regional features and then trains an action classifier. A method based on tracking human hands and their interactions with objects is presented in Ni et al [20]. Hough forests for action recognition are proposed in Gall et al [21]. Although recognizing objects may be helpful, it may be difficult to detect in the context of fine-grained actions.

We also note that there have been some other deep learning architectures devised for action modeling such as 3D convolutional filters, recurrent neural networks, long-short term memory networks, and large scale video classification architectures. These architectures demand huge numbers of videos for effective training; however, these are mostly unavailable for fine-grained activity tasks and thus the feasibility of these architectures is yet to be confirmed.

Pooling has been a useful scheme for reducing the size of video representations, thereby enabling the applicability of efficient machine learning algorithms to these data transformation. Recently, a pooling technique maintaining the temporal order of the frames was proposed by Fernando et al [22]. A method fusing deep features with action trajectories in video was proposed by Wang et al [8]. Early and late fusion of CNN feature maps for action recognition was discussed in [23, 24]. Our proposed hashing pooling scheme is somewhat similar to the second-order and higher-order pooling approaches proposed in [17] and [25], which generate representations from low-level descriptors for the semantic segmentation of images and object recognition. Moreover, our hashing descriptor is inspired by the sequence compatibility kernel (SCK) descriptor that was introduced in [26] which pools the higher-order occurrences of feature maps from skeletal body joints for action recognition. In contrast, we use the frame-level feature vectors (the output of the fully connected layers) from the deep classifiers to generate our pooled descriptors. Therefore, the size of our pooled descriptors is a function of the number of action classes. Moreover, unlike SCK, which uses pose skeletons, we use raw action videos directly. Our work is also different from works such as [27, 28] in which tensor descriptors were proposed for hand-crafted features. In this paper, we show how CNNs could benefit from higher-order pooling for the application of fine-grained action recognition.

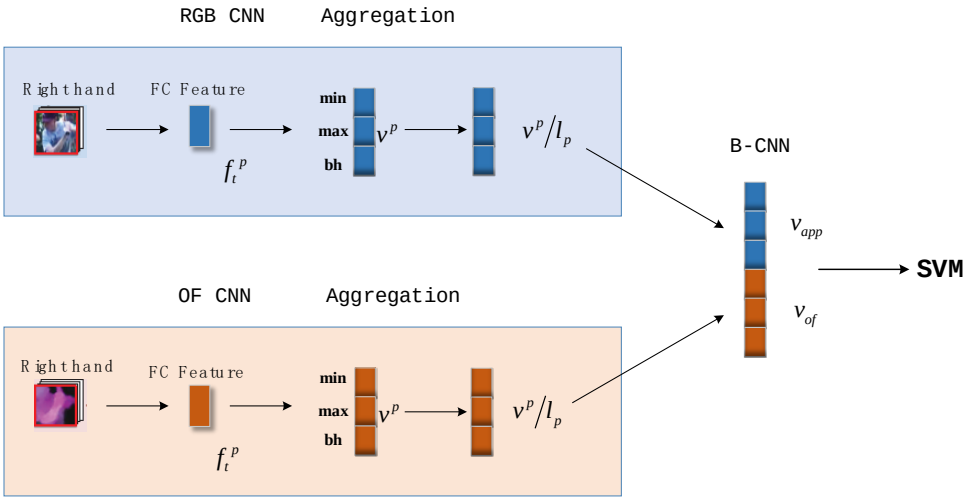

Fig. 2. The process of action recognition

3. B-CNN: Binary-hashing CNN feature

We assume that an action sequence contains vast amounts of information from many sub-actions. However, we believe that the key sub-actions are essential for action recognition. For example, the method in [12] depends on the appearance-based and motion-based CNN descriptors extracted from each video frame, which are aggregated over the timeline to form the video descriptor. These methods give equal weights to each video frame. However, we assume that the key sub-action carries the most important information. Therefore, we want to add the key sub-action information as a significant supplement. Consequently, we choose a key frame sequence that is a sub-set of the original video sequence. We proceed to extract the CNN features from this sequence and perform a feature comparison to construct the binary hashing feature. **Fig. 2** provides an overview. The details are described below.

To construct the hashing feature, we first compute the optical flow [29] for every consecutive pair of frames. Following the process described in [30], the values of the motion field are transformed into the interval [0, 255] by $\tilde{v}_{x|y} = 16 \times v_{x|y} + 128$. The values outside the interval are truncated. We save the transformed flow maps as images with three channels according to the motion and the flow intensity (shown in **Fig. 3**).

(a) (b) (c)

Fig. 3. (a),(b): a pair of consecutive video frames (c) transformed flow map

Given a video frame sequence and the corresponding joint positions with different body parts, we crop the image patches for the left hand, right hand, upper body, and full body from the RGB(appearance) and flow(motion) images. As **Fig. 2** shows, to distinguish between the appearance and motion patches, we use two distinct convolutional neural networks with an architecture similar to [31]. Hence, both networks have 5 equal convolutional and 3 fully connected layers. Each patch is adjusted to 224×224 pixels to fit the first layer in the network. The outputs of the second fully connected layer that are denoted as the FC features are used for the frame descriptor (f_t^p). For the RGB patches we use the publicly available "VGG-f " network from [32], which has been pre-trained using the ImageNet LSVRC 2012 dataset [33]. For flow patches, we use the motion network provided by [30], which has been pre-trained for the action recognition task on the UCF101 dataset [34]. Afterwards, we obtain the FC feature (f_t^p) as the frame descriptor from two convolutional neural networks.

Given the descriptors f_t^p for each body part p and the original video F with T frames, we can select h frames from the original video with its corresponding descriptors as Eq.(1). The proper choice of hashing size h may have an effect on the recognition performance as Eq.(2). We will discuss this in Section 4. After the key frames and their features are selected from the original video F, we then proceed with the comparison of two adjacent frames.

$$\begin{cases} [f_{t_1}, f_{t_2}, \ldots, f_{t_h}] \subseteq F, 2 \le h \le T \\ F = [f_1, f_2, \ldots, f_T] \end{cases} \tag{1}$$

$$\begin{cases} f_{t_i} = f_{1+(i-1)\times s}, i \in \{1, 2, \ldots, h\} \\ s = \left\lceil \frac{|F|}{h} \right\rceil \end{cases} \tag{2}$$

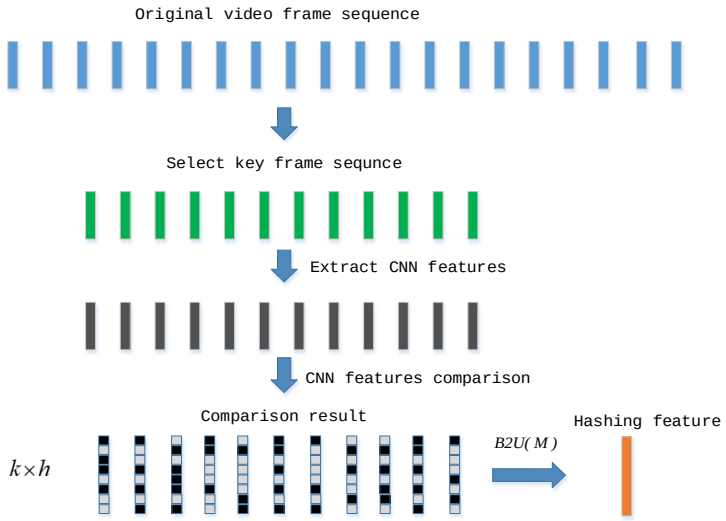

Fig. 4. The process of B-CNN feature.

Fig. 4 shows that we compare the adjacent descriptors with each dimension $k = 4096$ and obtain a binary value matrix $M_{k\times h}$. For the row vector $\vec{x}$ $[x_{h-1}, x_{h-2}, ..., x_0]$ in each row of *M*, we use Eq.(3). As Eq.(3) shows, the function $B2U_w$ maps a sequence of binary values of length w to a nonnegative integer, which denotes the variation between the video frames. Finally, we obtain the descriptor v_{hash}^p for the entire video *F* of part *p*.

$$B2U_w(\vec{x}) = \sum_{i=0}^{w-1} x_i \times 2^i \tag{3}$$

Based on [13], we also consider the max and min aggregation by computing the maximum and minimum values for each descriptor dimension *i* in *T* video frames as

$$M_i = \max_{1\le t\le T} f_t^p(\mathrm{i}), \mathrm{m}_i = \min_{1\le t\le T} f_t^p(\mathrm{i}) \tag{4}$$

The static video descriptor for part *p* is defined by the concatenation of the time-aggregated frame descriptor as Eq.(5).

$$v_{stat}^p = [M_1, ..., M_k, m_1, ..., m_k] \tag{5}$$

The dynamic video descriptor v_{dyn}^p comes from the temporal frame difference as $\Delta f_t^p = f_{t+\Delta t}^p - f_t^p$ for $\Delta_t = 4$ frame, and $\Delta_{M_i}, \Delta_{m_i}$ is computed as Eq.(6).

$$v_{dyn}^p = [\Delta_{M_1}, ..., \Delta_{M_k}, \Delta_{m_1}, ..., \Delta_{m_k}] \tag{6}$$

Finally, the static and dynamic video descriptors for the appearance and motion for all parts are concatenated to obtain a global video representation *p* for the video. To make this representation invariant to the number of extracted different descriptors, the concatenated result is further normalized by certain methods. Generally, there are two common normalization techniques:

- L_1 normalization. In L_1 normalization [35], the feature p is divided by its L_1 norm:

$$\mathbf{p} = \mathbf{p}/\|\mathbf{p}\|_1$$

- L_2 normalization. In L_2 normalization [4], the feature p is divided by its L_2 norm:

$$\mathbf{p} = \mathbf{p}/\|\mathbf{p}\|_2$$

In addition, we define and use a fusion normalization method as Eq.(7).

$$p = p/(\|p\|_1 + \beta \cdot \|p\|_2) \tag{7}$$

In Section 4, we evaluate the effect of different hashing sizes *h*, different normalization techniques and different fusion parameters β for the action recognition performance.

4. Experimental Results and Analysis

This section provides experimental evidence of the usefulness of our proposed hashing scheme and fusion-normalization for action recognition. We verify this task using two popular benchmark datasets, the MPII-Cooking Activities and the JHMDB dataset.

4. 1 Datasets

The MPII Cooking Activities Dataset [5] consists of a series of high-resolution videos of human actions occurring in a kitchen with the same activity background. Some of the actions are very similar. The dataset consists of videos of people cooking various dishes, slicing foods, washing their hands and washing objects. Each video contains a single person cooking a dish, and there are 12 such videos in the dataset in general. There are 64 distinct activities spread across 3748 video clips and one background activity (1861 clips). The activities range from coarse subject motions, such as moving objects, opening the refrigerator, etc., to fine-grained actions such as peeling, slicing, cutting, etc.

The JHMDB Dataset [36] is a subset of the larger HMDB dataset [37]. It contains 21 categories involving a single person engaging in an action such as brushing hair, sitting, standing, walking, waving, etc. The video clips only include a short duration of an action. Each video clip contains 15-40 frames and 36-55 clips per action. There are a total of 928 video clips containing 31838 annotated frames. There are 3 training/testing splits for the JHMDB dataset, and the evaluation averages the results over these splits.

4. 2 Evaluation Methods

Following the standard protocols, we use the mean average precision over 7-fold cross-validation on the MPII-Cooking dataset. Other datasets use the mean average accuracy of 3-splits. For the former, we use the evaluation code published with the dataset.

4. 3 Preprocessing

The original MPII cooking videos have very high resolution. While the activities occur only in specific parts of the scene, we use the difference of a frame to estimate a window of the scene to localize the action. Specifically, for each sequence, first we convert the frames to their half sizes. This is followed by frame-differencing, erosion, filtering, and connected component labeling. This constructs a binary image for each frame, which is then combined through the sequence, and a binary mask is generated for the whole sequence.

We use the largest bounding box containing all the connected components in this binary mask as the region of the action, and then crop the video to this box. To compute the optical flow, we use the Brox implementation [29]. Each flow image is rescaled to 0-255 and saved as a PNG image for storage efficiency as described in [12]. For the JHMDB dataset, the frames are already low resolution. Thus, we directly use them in the CNN after resizing them to the expected input sizes.

4. 4 Parameter Learning

As is obvious from Eq.(2) and Eq.(7) there are a few hyper-parameters associated with the binary hashing descriptor. In this subsection, we systematically analyze the effect of these parameters on the overall classification performance of the descriptor. For this purpose, we use the mean average accuracy over 3 training/testing splits for JHMDB. Specifically, we explore the effect of the changes on (i) different normalization methods, (ii) the factor β of the fusion normalization and (iii) hashing size *h*. In **Fig. 6**, we plot the average classifier accuracy for each of these cases. For (i) and (iii), we compare the hashing size of [6, 7, 8, 9, 10]. Under L_1 normalization, L_2 normalization and $L_1+\beta \cdot L_2$ normalization fusion with fixed $\beta = 0.03$. We compare the PCNN and the binary hashing feature, and combined the two methods for the JHMDB dataset. The result shows that the binary hashing feature is indispensable as [38] concluded, and motion information is essential for action recognition.

For (i) we compared the recognition accuracy using different normalization methods. As **Fig. 5** shows, there is no difference in choosing the normalization method when using only the hashing feature. Their performances are all worse than the combination of the P-CNN and hashing features.

However, the different normalization has a different effect when we choose feature fusion. Like **Fig. 6** shows, using L_1 normalization is slightly better than L_2 normalization. We also noted that the accuracy of fusion normalization is significantly higher than either normalization alone.

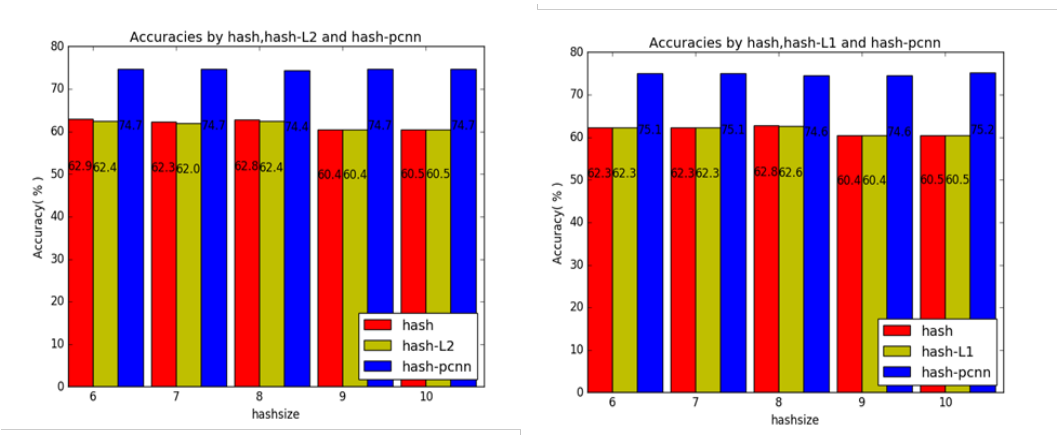

Fig. 5. Comparison among different normalization methods

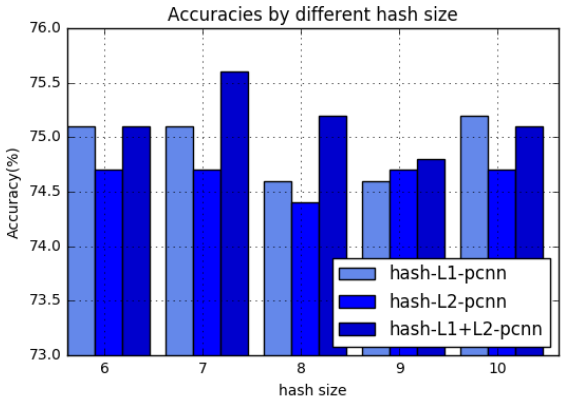

Fig. 6. Comparison among different hashing size

In fact, the normalization method is related to the kernel used in the final classifier. In our case of the linear SVM, the kernel is $k(x,y)=x^T y$. The choice of L_1 normalization can ensure two things: (i) $k(\mathrm{x},\mathrm{x})=\mathrm{const}$, and (ii) $k(\mathrm{x},\mathrm{x})<\mathrm{k}(\mathrm{x},\mathrm{y})$ feature selection in sparse feature spaces. This can guarantee a simple consistency criterion by interpreting $k(\mathrm{x},\mathrm{y})$ as a similarity score, which should be the most similar point to it. For (iii), as **Fig. 7** shows, as the rate of the fusion parameter β increases, the recognition accuracy increases as well. However, beyond a certain rate, the accuracy starts dropping. Meanwhile, we note that under almost every β setting, the accuracy with hashing size $h=7$ beats the other settings, and the accuracy with the hashing size $h=10$ is robust but is not the best. This is perhaps due to

the hashing size variation. Note that the JHMDB sequences contain approximately 30-40 frames per sequence. When the hashing size increases, the time difference between each pair of adjacent frames become shorter and the action variation becomes more prominent.

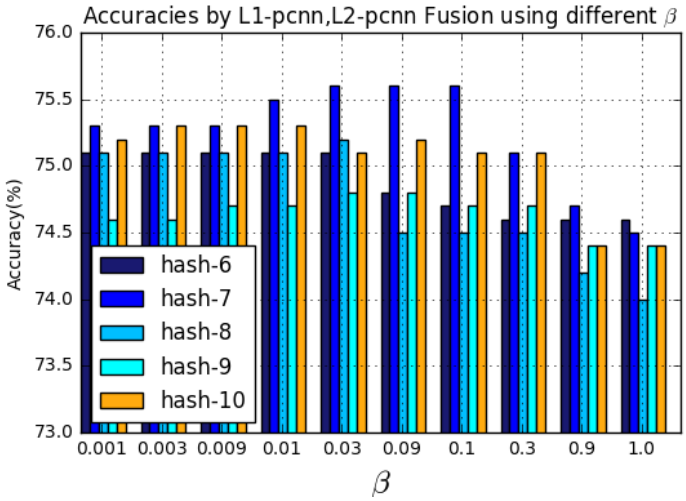

Fig. 7. Comparison among different normalization factor

4. 5 Results and Comparisons to the State of the Art

In this subsection, we provide full experimental comparisons of the two datasets. Our main goal is to analyze the usefulness of hashing to capture the motion information for action recognition.

In **Table 1**, we compared the results of original Hashing feature and feature fusion with the PCNN feature under different normalization methods. In **Table 2** and **Table 3**, we compared the binary hashing descriptors to the state-of-the-art dense trajectory features [33] encoded by Fisher vectors [22] and PCNN [6] results on these two datasets.

Table 1. Comparison of different normalization methods with hashing features and PCNN features for the JHMDB and MPII Cooking dataset (% accuracy).

Method	JHMDB	MPII Cooking
PCNN	74. 6	62. 3
Hashing	62. 9	57. 2
P+L1-Hashing	75. 1	63. 5
P+L2-Hashing	74. 7	62. 8
P+L1, L2 Hashing Fusion	**75. 6**	**63. 8**

Table 2. MPII Cooking Activities dataset (7-splits)

Algorithm	mAP(%)
IDT+FV ICCV'13	67. 6
PCNN ICCV'15	62. 3
Our B-CNN	**63. 8**

Table 3. JHMDB dataset (3-splits)

Algorithm	Avg. Acc. (%)
IDT+FV ICCV'13	65. 9
PCNN ICCV'15	74. 6
Our B-CNN	**75. 6**

For the JHMDB dataset, we use hashing size $h = 7$ for the classifier scores, and $\beta = 0.03$ for the feature fusion. The BCNN method improves the performance from 74.6% to 75.6%. We use the same setup for the MPII Cooking dataset except that we use the hashing size $h = 12$. The fusion normalization factor β is set to 0.3. The BCNN method improves the performance from 62.3% to 63.8%. As is clear, although the binary hashing feature by itself is not superior to other methods, when the PCNN, binary hashing, L_1 normalization and L_2 normalization are combined, it demonstrates significant promise.

5. Conclusion

In this paper, we presented a method for the higher-order pooling of CNN features for action recognition in videos. We showed how to use binary hashing to generate a higher-order descriptor that can capture motion information from a video. Our experimental analysis of two standard fine-grained action datasets clearly demonstrates that using the binary hashing feature method and fusion normalization is beneficial for the task and leads to state-of-the-art performance. A promising direction for future work is to adapt region- based CNNs [39] for each B-CNN part by fine-tuning networks for corresponding video frame areas. Another interesting direction is to model the temporal evolution of frames using RNNs [40].

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 61272195, U1713213 and U1401252), the National Key Research program of China (2016YFC1000307-3) and the Chongqing Outstanding Youth Fund (cstc2014jcyjjq40001). The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

[1] Aggarwal J K, Ryoo MS, "Human activity analysis: a review," *ACM Comput Surv* 43(3):1–43, 2011. Article (CrossRef Link)

[2] Peng X, Wang L, Wang X, Qiao Y, "Bag of visual words and fusion methods for action recognition Comprehensive study and good practice," *Computer Vision and Image Understanding*, 2016. Article (CrossRef Link)

[3] Peng X, Zou C, Qiao Y, Peng Q, "Action recognition with stacked fisher vectors," in *Proc. of Proceedings of European Conference on Computer Vision (ECCV)*, 581-595, 2014. Article (CrossRef Link)

[4] Herath S, Harandi M, and Porikli F. "Going deeper into action recognition: A survey," *Image and Vision Computing*,60:4 – 21, 2017. Article (CrossRef Link)

[5] Rohrbach M, Amin S, Andriluka M, Schiele B, "A database for fine grained activity detection of cooking activities," in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1194-1201, 2012. Article (CrossRef Link)

[6] Cherian A, Fernando B, Harandi M, and Gould S, "Generalized Rank Pooling for Activity Recognition," in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1581-1590, 2017. Article (CrossRef Link)

[7] 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. Article (CrossRef Link)

[8] Wang L, Qiao Y, Tang X, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 4305-4314, 2015. Article (CrossRef Link)

[9] Laptev I, "On space-time interest points," *Int J Comput Vis* 64(2–3):107–123, 2005. Article (CrossRef Link)

[10] Klaser A, Marszałek M., Schmid C, "A spatio-temporal descriptor based on 3d-gradients," in *Proc. of British Machine Vision Conference (BMVC)*, doi:10.5244/C.22.99, 2008. Article (CrossRef Link)

[11] Wang H, Schmid C., "Action recognition with improved trajectories," in *Proc. of Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3551-3558, 2013. Article (CrossRef Link)

[12] Simonyan K, Zisserman A, "Two-stream convolutional networks for action recognition in videos," in *Proc. of Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, 568-576, 2014.

[13] Chéron G, Laptev I, Schmid C, "P-CNN: Pose-based cnn features for action recognition," in *Proc. of Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3218-3226, 2015. Article (CrossRef Link)

[14] Du W, Wang Y, Qiao Y, “Rpan: An end-to-end recurrent pose-attention network for action recognition in videos,” in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3725-3734, 2017. Article (CrossRef Link)

[15] Liu M, Liu H, and Chen C. “Enhanced skeleton visualization for view invariant human action recognition,” *Pattern Recognition*, (68):346–362, 2017. Article (CrossRef Link)

[16] Tarn D, Bourdev L, Fergus R, Torresani L,Paluri M, “Learning spatiotemporal features with 3d convolutional networks,” in *Proc. of Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4489-4497, 2015. Article (CrossRef Link)

[17] Koniusz P, Yan F, Gosselin PH, Mikolajczyk K, “Higher-order occurrence pooling for bags-of-words: Visual concept detection,” *IEEE Trans Pattern Anal Mach Intell* 39(2), 313-326, 2017. Article (CrossRef Link)

[18] Cherian A, Mairal J, Alahari K, Schmid C, “Mixing body-part sequences for human pose estimation,” in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2353-2360, 2014. Article (CrossRef Link)

[19] Zhou Y, Ni B, Hong R, Wang M, Tian Q, “Interaction part mining: A mid-level approach for fine-grained action recognition,” in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 3323-3331, 2015. Article (CrossRef Link)

[20] Ni B, Paramathayalan VR, Moulin P, “Multiple granularity analysis for fine-grained action detection,” in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 756-763, 2014. Article (CrossRef Link)

[21] Gall J, Yao A, Razavi N, Van Gool L, Lempitsky V, “Hough Forests for Object Detection, Tracking, and Action Recognition,” *IEEE Trans Pattern Anal Mach Intell* 33:2188-2202, 2011. Article (CrossRef Link)

[22] Fernando B, Gavves E, Oramas J, Ghodrati A, Tuytelaars T, “Modeling video evolution for action recognition,” in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5378-5387, 2015. Article (CrossRef Link)

[23] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L, “Large-scale video classification with convolutional neural networks,” in *Proc. of Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 1725-1732, 2014. Article (CrossRef Link)

[24] Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G, “Beyond short snippets: Deep networks for video classification,” in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4694-4702,2015. Article (CrossRef Link)

[25] Carreira J, Caseiro R, Batista J, Sminchisescu C, “Semantic segmentation with second-order pooling,” in *Proc. of Proceedings of European Conference on Computer Vision (ECCV)*, 430-443, 2012. Article (CrossRef Link)

[26] Koniusz P, Cherian A, Porikli F, "Tensor representations via kernel linearization for action recognition from 3D skeletons," in *Proc. of Proceedings of European Conference on Computer Vision (ECCV)*, 37-53, 2016. Article (CrossRef Link)

[27] Koniusz P, Cherian A, "Sparse coding for third-order super-symmetric tensor descriptors with application to texture recognition," in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5395-5403, 2016. Article (CrossRef Link)

[28] Vasilescu M, Terzopoulos D, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. of Proceedings of European Conference on Computer Vision (ECCV)*, 447-460, 2002. Article (CrossRef Link)

[29] Brox T, Bruhn A, Papenberg N, Weickert J, "High accuracy optical flow estimation based on a theory for warping," in *Proc. of Proceedings of European Conference on Computer Vision (ECCV)*, 25-36, 2004. Article (CrossRef Link)

[30] Gkioxari G, Malik J, "Finding action tubes," in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 759-768, 2015. Article (CrossRef Link)

[31] Alex Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, 568-576, 2012. Article (CrossRef Link)

[32] Chatfield K, Simonyan K, Vedaldi A, Zisserman A, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," *british machine vision conference*, 2014.

[33] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L, "Imagenet: A large-scale hierarchical image database," in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248-255, 2009. Article (CrossRef Link)

[34] Soomro K, Zamir AR, Shah M, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, abs/1212.0402, 2012.

[35] Yang J, Yu K, Gong Y, Huang T, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1794-1801, 2009. Article (CrossRef Link)

[36] Jhuang H, Gall J, Zuffi S, Schmid C, Black MJ., "Towards understanding action recognition," in *Proc. of Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3192-3199,2013. Article (CrossRef Link)

[37] Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T, "HMDB: a large video database for human motion recognition," in *Proc. of Proceedings of the IEEE Conference on Computer Vision (ICCV)*, 2556-2563, 2011. Article (CrossRef Link)

[38] Oneata D, Verbeek J, Schmid C, "Action and event recognition with fisher vectors on a compact feature set," in *Proc. of Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1817-1824, 2013. Article (CrossRef Link)

[39] Peng X, Schmid C, "Multi-region two-stream R-CNN for action detection," in *Proc. of Proceedings of European Conference on Computer Vision (ECCV)*, 744-759, 2016. Article (CrossRef Link)

[40] Zhu W, Lan C, Xing J, Zeng W, Li Y, Shen L, Xie X, "Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks," in *Proc. of Proceedings of Thirtieth AAAI Conference on Artificial Intelligence*, 3697-3703, 2016.

Weisheng Li graduated from School of Electronics & Mechanical Engineering at Xidian University, Xian, China in July 1997. He received M.S. degree and Ph.D. degree from School of Electronics & Mechanical Engineering and School of Computer Science & Technology at Xidian University in July 2000 and July 2004, respectively. Currently he is a professor of Chongqing University of Posts and Telecommunications. His research focuses on intelligent information processing and pattern recognition.

Chen Feng received his B.S. degree in Computer Science and Technology from Henan University, Kaifeng, China. He is currently a graduate student at the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, China. His research interests include action recognition, machine learning and its application.

Bin Xiao was born in 1982. He received his B.S. and M.S. degrees in Electrical Engineering from Shaanxi Normal University, Xi'an, China in 2004 and 2007, received his Ph. D. degree in computer science from Xidian University, Xi'An, China. He is now working as an associate professor at Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include image processing and pattern recognition

Yanquan Chen has received B.S. degree from Beijing University of Chemical Technology, Beijing, P.R. China. Currently, he is a M.S. candidate in computer technology, with the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications. His research interests include image processing and pattern recognition.