Unsupervised Video Hashing by Exploiting Spatio-Temporal Feature

Chao Ma, Yun Gu, Wei Liu, and Jie Yang*

Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China.
{sjtu_machao,geron762,liuwei.1989,jieyang}@sjtu.edu.cn

Abstract. Video hashing is a common solution for content-based video retrieval by encoding high-dimensional feature vectors into short binary codes. Videos not only have spatial structure inside each frame but also have temporal correlation structure between frames, while the latter has been largely neglected by many existing methods. Therefore, in this paper we propose to perform video hashing by incorporating the temporal structure as well as the conventional spatial structure. Specifically, the spatial features of videos are obtained by utilizing Convolutional Neural Network (CNN), and the temporal features are established via Long-Short Term Memory (LSTM). The proposed spatio-temporal feature learning framework can be applied to many existing unsupervised hashing methods such as Iterative Quantization (ITQ), Spectral Hashing (SH), and others. Experimental results on the UCF-101 dataset indicate that by simultaneously employing the temporal features and spatial features, our hashing method is able to significantly improve the performance of existing methods which only deploy the spatial feature.

Keywords: Video Hashing, Unsupervised Method, Spatio-temporal Feature

1 Introduction

Video retrieval is a very challenging task in the area of computer vision. Most of current video search engines rely on textual keyword matching rather than visual content-based retrieval. One of the bottlenecks for content-based search is the unaffordable computational cost when handling a large collection of video clips. Consequently, hashing is a popular method to solve this problem by encoding high-dimensional feature vectors into short binary codes, so that the hamming distance, which is very efficient to compute, can be used to represent the similarity between different videos. This has enabled significant efficiency gains in both storage and speed.

Recently, great achievements have been made on hashing by incorporating various machine learning techniques. These methods can be divided into three categories: unsupervised, semi-supervised, and supervised. Unsupervised hashing

* Corresponding author: Jie Yang. {jieyang@sjtu.edu.cn}

methods such as Spectral Hashing (SH) [17] mainly utilize data properties like distribution or manifold structure to design effective indexing schemes. Supervised methods such as deep neural network-based method [15] treat the design of hash functions as a special classification problem and utilize supervised (label) information in the training procedure. Some other supervised methods, *e.g.* supervised hashing with kernels [4], take into account the pairwise relationship of samples in the hash function learning procedure. Semi-supervised hashing methods [16] play a tradeoff between supervised information and data properties, which leverage semantic similarity using label data while remaining robust to overfitting.

Video hashing is different from image hashing because videos not only have the spatial structure within each frame but also have the temporal correlation between frames. On one hand, Convolutional Neural Network (CNN) can be used to learn spatial structure features, as CNN is effective in learning rich mid-level image descriptors. By utilizing the feature vectors generated by the seventh layer in the trained CNN, the method proposed by Krizhevsky *et al.* [13] achieved the state-of-the-art performance in image retrieval on ImageNet dataset [2].

On the other hand, there are also networks that perfectly learn the correlation between signal sequences. It is widely acknowledged that Recurrent Neural Network (RNN) models are "deep in time", which means that RNN is connectionist models that capture the dynamics of sequences via cycles in the nodes of network. However, a significant limitation of simple RNN models is the "vanishing gradient" effect, *i.e.* practically back propagating an error signal through a long-range temporal interval will become increasingly intractable. To handle the vanishing gradient problem, Hochriter *et al.* [9] introduced the Long-Short Term Memory (LSTM) model which resembled a standard RNN with a hidden layer. Each ordinary node in the hidden layer is replaced by a memory cell that contains a node with a self-connected recurrent edge of fixed weight. This ensures that the gradient can pass across a long-range temporal interval without vanishing or exploding.

Most of recent hashing methods [4, 15–17] generate binary codes for each sample independently but pay little attention to developing specific hash functions to index structured data like videos. Recently, Song *et al.* [11] proposed multiple feature based hashing for video near-duplicate detection. Cao *et al.* [1] proposed a submodular hashing framework to index videos. Although these methods achieve satisfactory performance to some extent, the specific temporal structure between video frames is neither considered nor encoded into the binary codes.

In this paper, we propose to perform video hashing by making use of not only the spatial structure within each frame but also the temporal correlation structure between frames. We construct a spatio-temporal feature learning framework by using CNN for spatial feature learning and LSTM for temporal feature learning. We apply our spatio-temporal feature learning framework to many unsupervised hashing methods including Iterative Quantization (ITQ) [7], Locality Sensitive Hashing (LSH) [6], PCA Hashing (PCAH) [7], Spectral Hashing (SH)

[17], Density Sensitive Hashing (DSH) [12], and Spherical Hashing (SpH) [8]. We use UCF-101 dataset [14] to compare our hashing method and the existing algorithms that only deploy the spatial features, and the results reveal that our method is significantly superior to the existing methodologies.

The rest of our paper is organized as follows: The details of our approach are described in Section 2. Our approach is empirically evaluated on the UCF-101 dataset in Section 3. Finally, we conclude the entire paper in Section 4.

2 Methodology

2.1 The Recurrent Neural Networks

Fig. 1(a) is a simple recurrent net with one input unit, one output unit and one recurrent hidden unit. Such recurrent net can learn complex temporal dynamics by mapping input sequences to a sequence of hidden states. The hidden states are then mapped to the output via the following recurrence equations:

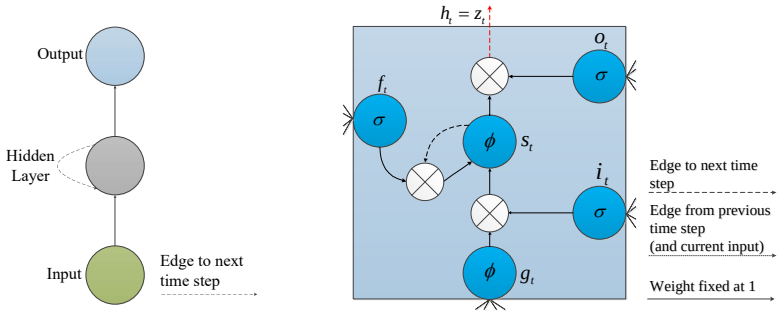
$$\begin{aligned} h_t &= g(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \\ z_t &= g(W_{hz}h_t + b_z) \end{aligned} \tag{1}$$

where $g(\cdot)$ is an element-wise non-linearity function, such as sigmoid or hyperbolic tangent, x_t is the input, $h_t \in \mathbb{R}^N$ denotes the hidden state with N hidden units, and z_t is the output at time t. The weight matrices W_{ij} and biases b_j are the parameters to be learned.

As mentioned in the introduction, although RNN has been proven to be successful on several tasks, a significant drawback of simple RNN models is the vanishing gradient problem. This makes it difficult to train RNN to learn long-term dynamics. As a result, Long-Short Term Memory (LSTM) model provides a solution by incorporating memory units that allow the network to decide whether to forget the previous hidden states or to update the hidden states according to the new information. We use the LSTM unit as described in [5] (Fig. 1(b)), which was derived from the LSTM initially proposed in [9]. The formal definition of LSTM with forget gates is formulated as follows:

$$\begin{aligned} g_t &= \phi(W_{gx}x_t + W_{gh}h_{t-1} + b_g) \\ i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \\ f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \\ o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \\ s_t &= g_t \odot i_t + s_{t-1} \odot f_t \\ h_t &= \phi(s_t) \odot o_t \end{aligned} \tag{2}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid non-linearity function which maps real-valued inputs into the interval [0, 1], $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$ is the hyperbolic tangent nonlinearity which maps its inputs into the interval [-1, 1], $\odot$ donates element-wise product.



(a) (b)

Fig. 1. Recurrent Neural Networks: (a) A simple recurrent net. (b) LSTM memory cell used in this paper.

The advantages of LSTM for modeling sequential data in computer vision are twofold. First, when integrated with current vision systems, the parameters in the LSTM model can be easily fine-tuned in an end-to-end way. Second, LSTM models are not limited to fixed length inputs. It is able to model the sequential data with varying length such as videos.

2.2 Obtaining Binary Codes of Videos

As CNN has shown impressive performance for spatial feature learning in various tasks, such as image classification [13], we thus adopt CNN to learn the spatial features in our spatio-temporal feature learning framework. Besides, we use the LSTM model to learn the temporal features between frames. Fig. 2 shows the proposed method. Our method includes two main components. The first component involves the supervised pre-training to learn the rich mid-level video representation features. We use the CNN+LSTM model proposed by Donahue *et al.* [3] in the Caffe library [10]. The model is then trained on the UCF-101 dataset [14]. The second component is the unsupervised hashing which exploits the spatio-temporal features obtained by the first component. The entire procedure for obtaining binary codes of spatio-temporal features is detailed as follows.

As illustrated in Fig. 2, we pass the visual input v_{it} (the t-th frame from the i-th video) through a few convolutional layers. The architecture used for each CNN layer is similar to the one proposed in [3]. The initial weights of CNN are set to the same values as trained on the ImageNet dataset [2]. The weights are then fine-tuned when the LSTM layer is added. The output of the fc_6 layer is chosen as the output of the CNN framework, which has 4096-dimensional feature. Rectified Linear Units (ReLU) and dropout layers are used to avoid the overfitting problem.

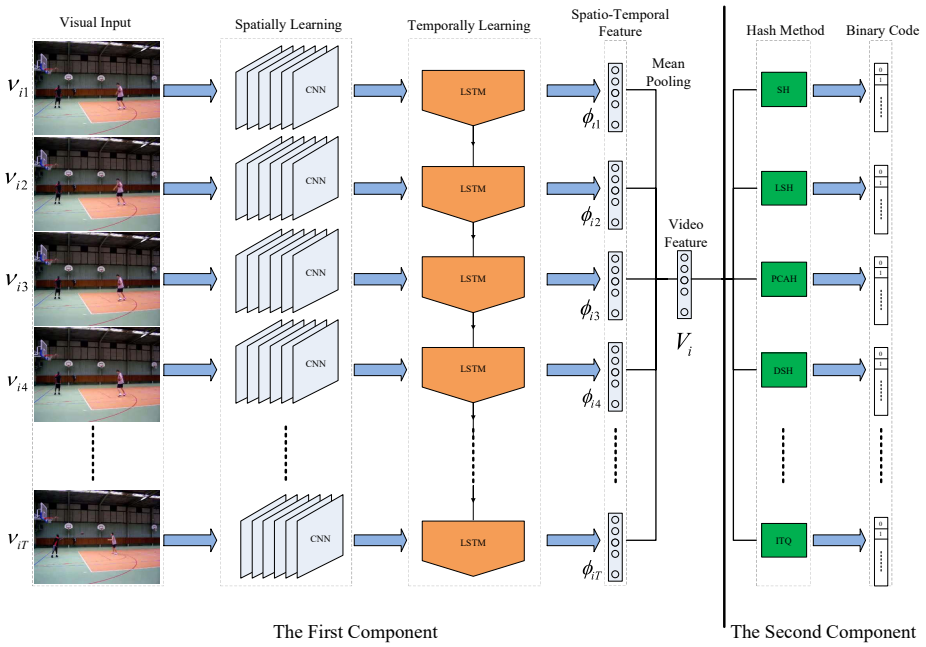


Fig. 2. The proposed video hashing framework consists of two main components. The first component involves the supervised per-training based on the UCF-101 dataset to learn the spatio-temporal features. The second component is unsupervised hashing which exploits the spatio-temporal features generated by the first component.

After modeling the spatial features with CNN, we pass the learned spatial features to LSTM layer. The LSTM layer is used to model the temporal correlation structure between frames in videos. Similar to the strategy proposed in [3], we use only one LSTM layer of which the output is a 256-dimensional vector. The reason why we choose only one LSTM layer is that our model requires to deal with sequential input while the output is a fixed-length vector, therefore other LSTM layers are not necessarily needed.

After all the frames have been processed, we obtain a vector representation sequence $\{\phi_{i1}, \phi_{i2}, ..., \phi_{iT}\}$. The spatio-temporal representation of each video is then calculated by:

$$V_i = \frac{1}{T} \sum_{t=1}^{T} \phi_{it} \tag{3}$$

where T is the number of frames in the i-th video.

After obtaining the feature set $V = \{V_1, V_2, \ldots, V_m\} \in \mathbb{R}^d$ (m is the number of videos), the next step for hashing is to look for a group of appropriate hash functions $h : \mathbb{R}^l \longmapsto \{1, -1\}^1$ with each of them accounting for generating of a single hash bit. Many unsupervised hashing methods can be used here such as Iterative Quantization (ITQ) [7], Spectral Hashing (SH) [17], and others.

3 Experiments

In this section, we apply our spatio-temporal feature learning framework to many existing unsupervised hashing methods including Iterative Quantization (ITQ) [7], Locality Sensitive Hashing (LSH) [6], PCA Hashing (PCAH) [7], Spectral Hashing (SH) [17], Density Sensitive Hashing (DSH) [12], and Spherical Hashing (SpH) [8]. By testing the proposed framework on the UCF-101 dataset, we compare the hashing methods with our spatio-temporal feature learning framework and the ones that only use spatial features which are learnt by CNN. We start by introducing the dataset and evaluation metrics, and then present the comparison results of our method with existing approaches on the UCF-101 dataset.

3.1 Dataset

UCF-101 Dataset [14] consists of 101 action classes, over $13k$ clips and 27 hours of video data. This database consists of realistic user-uploaded videos containing camera motion and cluttered background. In our experiments, we select 9,537 videos as the training data, and the remaining 3,783 videos are adopted for testing.

3.2 Evaluation Metrics

The performance of video retrieval is evaluated by Mean Average Precision (MAP). For a query q, the average precision is defined as:

$$AP(q) = \frac{1}{L_q} \sum_{r=1}^{R} P_q(r)\delta_q(r) \tag{4}$$

where L_q is the number of groundtruth neighbors in the retrieval list. $P_q(r)$ is the precision of the top r retrieved results, and $\delta_q(r) = 1$ if the r-th result is the true neighbor and 0 otherwise.

3.3 Results

To show the effectiveness of the proposed spatio-temporal feature learning framework, we first apply it to the unsupervised hashing methods mentioned above and see the results. Then we compare the results with the corresponding methods that only use the spatial features learnt by CNN. Fig. 3(a) shows the comparison results. It is clear that the methods using the proposed spatio-temporal features obtain significantly larger MAP than the ones that only use spatial features. This is because our spatio-temporal feature learning framework fully takes advantages of not only the spatial structure but also the temporal correlation of video data. The proposed spatio-temporal feature learning framework has a better representation of the video than spatial feature representation.

To further validate the effectiveness of the proposed method using binary codes of shorter bits, we perform the corresponding experiments by using the

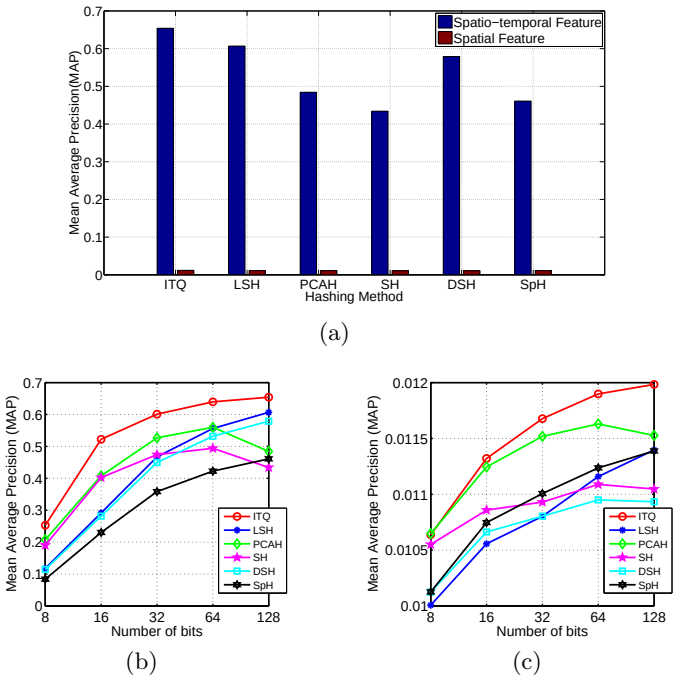


Fig. 3. Results of six unsupervised hashing methods on UCF-101. (a) The performance comparison between using spatio-temporal features and only using the spatial features. The comparison is performed on the 128-bit binary codes. (b) MAP results of the methods using the spatio-temporal features proposed in this paper. (c) MAP results of the methods only using spatial features learnt by CNN.

binary hashing codes of 8 bits, 16 bits, 32 bits, and 64 bits, respectively. The corresponding results are shown in Fig. 3(b) and Fig. 3(c). It can be clearly observed that the MAP results of both settings (*c.f.* using the proposed spatio-temporal features and only the spatial features) increase as the length of hashing binary codes become longer. However, the methods that exploit the proposed spatio-temporal features still outperform the ones that only use spatial features with a noticeable gain regarding MAP.

4 Conclusion

In this paper, we combine CNN and LSTM unit into a unified framework, which is both spatially and temporally deep, for video hashing. Due to the utilization of spatio-temporal features, the created binary codes are more representative for video retrieval, and thus encouraging performances can be achieved. Experimental results on UCF-101 dataset well demonstrate the feasibility and effectiveness of the proposed method.

Reference

1. Cao, L., Li, Z., Mu, Y., Chang, S.F.: Submodular video hashing: a unified framework towards video pooling and indexing. In: Proceedings of the 20th ACM international conference on Multimedia. pp. 299–308. ACM (2012)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009)
3. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Computer Vision and Pattern Recognition, 2015. CVPR 2015. IEEE Conference on. pp. 2625–2634 (2015)
4. Douze, M., Jégou, H., Schmid, C.: An image-based approach to video copy detection with spatio-temporal post-filtering. Multimedia, IEEE Transactions on 12(4), 257–266 (2010)
5. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with lstm. Neural computation 12(10), 2451–2471 (2000)
6. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: VLDB. vol. 99, pp. 518–529 (1999)
7. Gong, Y., Lazebnik, S.: Iterative quantization: A procrustean approach to learning binary codes. In: Computer Vision and Pattern Recognition, 2011. CVPR 2011. IEEE Conference on. pp. 817–824. IEEE (2011)
8. Heo, J.P., Lee, Y., He, J., Chang, S.F., Yoon, S.E.: Spherical hashing. In: Computer Vision and Pattern Recognition, 2012. CVPR 2012. IEEE Conference on. pp. 2957–2964. IEEE (2012)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
10. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
11. Jiang, Y.G., Ye, G., Chang, S.F., Ellis, D., Loui, A.C.: Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In: Proceedings of the 1st ACM International Conference on Multimedia Retrieval. p. 29. ACM (2011)
12. Jin, Z., Li, C., Lin, Y., Cai, D.: Density sensitive hashing. Cybernetics, IEEE Transactions on 44(8), 1362–1371 (2014)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
14. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
15. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large image databases for recognition. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
16. Wang, J., Kumar, S., Chang, S.F.: Semi-supervised hashing for large-scale search. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(12), 2393–2406 (2012)
17. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 21, pp. 1753–1760. Curran Associates, Inc. (2009)