2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

DLSTM Approach to Video Modeling with Hashing for Large-Scale Video Retrieval

Naifan Zhuang
Department of Computer Science
University of Central Florida
Orlando, Florida 32816
zhuangnaifan@knights.ucf.edu

Jun Ye
Department of Computer Science
University of Central Florida
Orlando, Florida 32816
jye@cs.ucf.edu

Kien A. Hua
Department of Computer Science
University of Central Florida
Orlando, Florida 32816
kienhua@cs.ucf.edu

***Abstract*—Although Query-by-Example techniques based on Euclidean distance in a multidimensional feature space have proved to be effective for image databases, this approach cannot be effectively applied to video since the number of dimensions would be massive due to the richness and complexity of video data. The above issue has been addressed in two recent solutions, namely Deterministic Quantization (DQ) and Dynamic Temporal Quantization (DTQ). DQ divides the video into equal segments and extracts a visual feature vector for each segment. The bag-of-word feature is then encoded by hashing to facilitate approximate nearest neighbor search using Hamming distance. One weakness of this approach is the deterministic segmentation of video data. DTQ improves on this by using dynamic video segmentation to obtain varied-length video segments. As a result, feature vectors extracted from these video segments can better capture the semantic content of the video. To support very large video databases, it is desirable to minimize the number of segments in order to keep the size of the feature representation as small as possible. We achieve this by using only one video segment (i.e., no video data segmentation is even necessary) with even better retrieval performance. Our scheme models video using differential long short-term memory (DLSTM) recurrent neural networks and obtains a highly compact fixed-size feature representation with the output of hidden states of the DLSTM. Each of these features are further compressed by hashing them into binary bits via quantization. Experimental results based on two public data sets, UCF101 and MSRActionPairs, indicate that the proposed video modeling technique outperforms DTQ by a significant margin.**

I. Introduction

As video capturing devices become more and more ubiquitous, huge amounts of videos are uploaded to social networks each day and are searched and consumed by billions of people around the world. In the face of the big data era, there is an urgent demand for indexing and retrieving these ever-increasing video stores in an accurate and efficient way.

Although video retrieval using natural language description or keywords is highly desirable, the majority of videos lack tag information. Due to the semantic gap between low level visual features and high level semantic information, effective automatic video tagging techniques are still immature from being applied. Most existing video retrieval techniques are built upon physical feature-based modeling, in which a video is represented and indexed by visual features that can be extracted automatically. In this environment, Query by Example (QBE) is a common strategy for retrieving similar videos from a database.

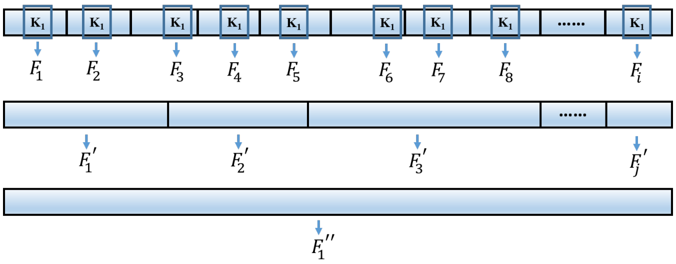

Fig. 1. An illustration of video modeling strategies. The top row shows deterministic quantization. The middle row shows dynamical quantization. The bottom row shows the proposed modeling method.

One approach to QBE is to model each video clip using a feature vector. The video clips in the database can be viewed as points in the multidimensional feature space. To retrieve similar video clips, the feature vector is extracted from the query video and the video clips whose feature points are closest to the query point in the multidimensional Euclidean feature space are returned. Although this approach has been quite effective for image retrieval, its application to video retrieval has many disadvantages. The feature vector would be very large with many components due to the richness and complexity of video data. The vastness of the high-dimensional Euclidean space causes these techniques to perform poorly, a phenomenon known as the curse of high dimensionality.

To overcome the weaknesses of the aforementioned QBE approach based on Euclidean distances, QBE based on Hamming distance has been proposed [21], in which a video clip is divided into video segments of equal length, e.g., 1 second. For each segment, a key frame is determined and its SIFT visual feature is extracted as shown in the top illustration in Fig. 1. These features form the bag-of-word representation and can be encoded by hash functions learned through temporal consistency. This approach facilitates similarity computation based on approximate nearest neighbor (ANN) search using Hamming distance [9], which is much more efficient than the solutions based on Euclidean distance.

A weakness of the technique presented in [21] is the deterministic nature of the uniform video data segmentation

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

technique that does not take into consideration the semantic information in the video. This is addressed in [22] by using varied-length video segments as shown in the middle illustration in Fig. 1. These video segments are more consistent with the semantic content of the video clip, resulting in feature vectors that better capture the video information.

In order to allow the technique to scale up to support even very large video retrieval applications, it is highly desirable to substantially reduce the number of feature vectors used in [21] and [22], say to only 1 video segment. As shown in the bottom illustration in Fig. 1, video data segmentation is not needed. Ideally, this could be achieved with even better retrieval accuracy. We accomplish this task by modeling the video clip using differential long short-term memory (DLSTM) recurrent neural networks and obtaining a highly compact fixed-size feature representation using the output of the hidden states. Each of these features can then be hashed into binary bits for further compression. Our performance study based on two public data sets indicates that DLSTM video modeling is able to capture the most salient spatio-temporal patterns in the video. The experimental results show the superiority of this technique over the latest video modeling method presented in [22].

Without loss of generality, we assume the proposed video modeling technique with hashing is applied to the entire video clip. This technique is also applicable to larger videos consisting of many video story units. For these applications, a video story unit is the unit for video retrieval. The proposed technique can be applied to each story unit to facilitate QBE.

The remainder of this paper is organized as follows. We briefly review the related work in Section II. The proposed DLSTM-based video retrieval technique is introduced in Section III. We discuss the experimental results on the two public video data sets, UCF101 [14] and MSRActionPairs [11], in Section IV. Finally, we conclude this paper in Section V and talk about future work in Section VI.

II. Related Works

Hashing algorithms are widely adopted in the approximate nearest neighbor search problem [7]. There are plenty of works aimed at achieving a higher retrieval accuracy with a shorter code length. Existing hashing algorithms can be roughly classified into random-based methods and learning-based methods. Many of the random-based methods, such as LSH [7], encode the data by space-partitioning; other methods, such as Winner-Take-All Hashing (WTA) [20], explore the rank space and encode the ordinal relation between the features. The learning-based methods are normally data-dependent. Principle linear projections like PCA Hashing (PCAH) [18] and its rotation variant, Iterative Quantization (ITQ) [4], were proposed to minimize the quantization loss with respect to the original features. Supervised hashing methods, such as KSH [9], utilize the pairwise label-similarity and are capable of learning more discriminative hash codes. Recently, pointwise-based methods, such as Supervised Discrete Hashing (SDH) [13], have been proposed to decouple the optimization procedure from the hashing function and have achieved superior results.

All of the aforementioned methods are for static data such as images; hashing on sequential data like videos is relatively under-researched. A few existing video hashing methods, such as submod ular video hashing [2] and video hashing via structure learning [21], still perform the hashing on key frames rather than on the entire video. One of the great challenges for video hashing is to achieve a fixed-size feature representation that can capture the complex spatio-temporal dynamics of the target video. To this goal, different video modeling methods have been proposed. Dynamic Temporal Warping (DTW) [10] and temporal pyramid [19], [11] were proposed to model the temporal structure of videos. However, these methods suffer from temporal misalignment when computing the distance between two videos of different length. A dynamic temporal quantization algorithm was introduced in [22] to model the temporal structure of videos dynamically while preserving the temporal order. Other modeling approaches, such as Bag-of-Words (BoW) [16], have also been proposed to represent the videos by the histogram of visual words.

Long Short-Term Memory (LSTM) [6] was proposed to learn the dynamics of a long sequence. DLSTM [17], an upgraded version of LSTM, takes into consideration the impact of spatio-temporal dynamics and is better at learning salient patterns. Recently, there have been emerging studies on LSTM/DLSTM applications, such as speech recognition [5], multimodal translation [15] and action recognition [1]. However, LSTM/DLSTM has not been studied for video retrieval tasks.

III. DLSTM-based Video Modeling

In this section, we present the detailed algorithm for the DLSTM-based video modeling and discuss the rationale behind it. To make the paper self-contained, we first briefly introduce the Differential Long Short-Term Memory (DLSTM) Recurrent Neural Networks. Then, we introduce the objective function aiming to minimize the training loss defined on the pairwise label-similarity. We further demonstrate and analyze the optimization results. Finally, we introduce the fixed-size representation of the video from the DLSTM modeling for video retrieval.

A. *Differential Long Short-Term Memory*

Consider an input sequence $\{\mathbf{x}_t \in \mathbb{R}^M | t = 1, 2, ..., T\}$. A recurrent neural network (RNN) computes the hidden state sequences $\{\mathbf{h}_t \in \mathbb{R}^N | t = 1, 2, ..., T\}$ by

$$\mathbf{h}_t = tanh(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{hx}\mathbf{x}_t + \mathbf{b}_h), \tag{1}$$

where the hyperbolic tangent $tanh(\cdot)$ is an activation function in the range [-1, 1], $\mathbf{W}_{h*}$ are weighting matrices, and $\mathbf{b}_h$ is the bias vector.

For classification tasks, the hidden states will be mapped to a sequence $\{\mathbf{y}_t \in \mathbb{R}^K | t = 1, 2, ..., T\}$ by

$$\mathbf{y}_t = tanh(\mathbf{W}_{yh}\mathbf{h}_t + \mathbf{b}_y), \tag{2}$$

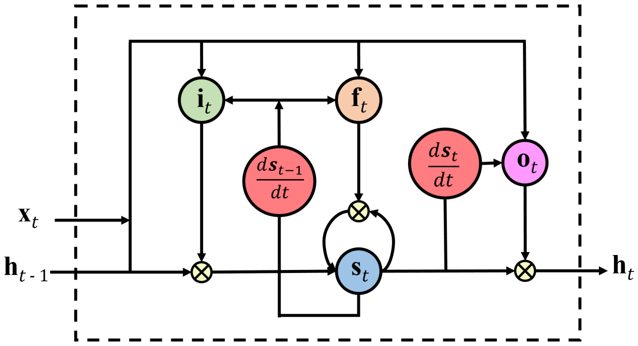

Fig. 2. Architecture of the DLSTM model at time t.

where each $\mathbf{y}_t$ represents a 1-of-K encoding of the confidence scores on K categories.

Due to exponential decay, traditional RNNs are limited in learning long-term sequences. Hochreiter *et al.* [6] designed Long Short-Term Memory (LSTM) to exploit the long-range dependency. According to recent study, the Derivative of States (DoS) in differential long short-term memory (DLSTM) [17] can explicitly model spatio-temporal structure and better learn salient patterns within. Fig. 2 shows the structure of DLSTM.

Replacing internal state with the DoS in the gate units, the DLSTM has the following updated equations:

(i) Input gate $\mathbf{i}_t$ regulates how much input information enters the memory cell to affect its internal state $\mathbf{s}_t$ at time t, which is defined as

$$\mathbf{i}_t = \sigma(\mathbf{W}_{id}\frac{d\mathbf{s}_{t-1}}{dt} + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{W}_{ix}\mathbf{x}_t + \mathbf{b}_i), \qquad (3)$$

where the sigmoid $\sigma(\cdot)$ is an activation function in the range [0,1].

(ii) Forget gate $\mathbf{f}_t$ gates the contribution of the the previous state $\mathbf{s}_{t-1}$ to the current state. It has the following recurrent form

$$\mathbf{f}_t = \sigma(\mathbf{W}_{fd}\frac{d\mathbf{s}_{t-1}}{dt} + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{W}_{fx}\mathbf{x}_t + \mathbf{b}_f). \qquad (4)$$

The internal state $\mathbf{s}_t$ of each memory cell can then be updated using the input and forget gate units, as shown below

$$\mathbf{s}_t = \mathbf{f}_t \odot \mathbf{s}_{t-1} + \mathbf{i}_t \odot \widetilde{\mathbf{s}}_t, \qquad (5)$$

where $\odot$ stands for element-wise product. Pre-state $\widetilde{\mathbf{s}}_t$ is defined as

$$\widetilde{\mathbf{s}}_t = tanh(\mathbf{W}_{sh}\mathbf{h}_{t-1} + \mathbf{W}_{sx}\mathbf{x}_t + \mathbf{b}_s).$$

(iii) Output gate $\mathbf{o}_t$ controls the information output from a memory cell, which can be expressed as

$$\mathbf{o}_t = \sigma(\mathbf{W}_{od}\frac{d\mathbf{s}_t}{dt} + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{W}_{ox}\mathbf{x}_t + \mathbf{b}_o). \qquad (6)$$

The hidden state of a memory cell, which contains the spatio-temporal information of previous frames, is output as

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{W}_{hs}\mathbf{s}_t + \mathbf{b}_h). \qquad (7)$$

By iteratively applying Eq. 5 and Eq. 7, DLSTM updates the internal state $\mathbf{s}_t$ and the output hidden state $\mathbf{h}_t$ over time.

As the DLSTM model is defined in the discrete-time domain, the derivative $\frac{ds_t}{dt}$ is then discretized as the difference of states by

$$\mathbf{d}_t \triangleq \frac{d\mathbf{s}_t}{dt} \doteq \mathbf{s}_t - \mathbf{s}_{t-1}. \qquad (8)$$

B. Objective and Pairwise-based Training

Previously, LSTM and DLSTM were mostly employed for recognition and classification tasks including speech recognition [5] and action recognition [1], *etc*. In these applications, the model parameters of the neural networks are learned through the pointwise fashion on the labeled data. In the context of video retrieval in the framework of approximate nearest neighbor search, we wish for the DLSTM to learn the most salient spatio-temporal patterns between similar video classes and to generate a highly compact representation of the original video which preserves the pairwise label-similarity.

To achieve this goal, we formulate an objective by leveraging the pairwise label-similarity. Specifically, consider two videos V_i and V_j with lengths T_i and T_j, respectively. The loss function can be defined as

$$\ell(i,j) = -log\frac{1}{1 + exp(\beta l_{ij}\|\mathbf{h}^i_{T_i} - \mathbf{h}^j_{T_j}\|_2)}, \qquad (9)$$

where $l_{ij} \in \{-1, +1\}$ denotes the label-similarity between V_i and V_j, with $+1$ indicating V_i and V_j are of the same label and -1 indicating otherwise. β is a normalizing factor. $\mathbf{h}^i_{T_i}$ and $\mathbf{h}^j_{T_j}$ denote the hidden state of V_i and V_j at the last time-step. Due to DLSTM's ability to capture long-term memory, the neural network accumulates increasingly richer information as it goes through the video. When the DLSTM reaches the last time-step, the hidden layer provides a semantic representation of the whole video. Thus, we train the DLSTM model based on the hidden state at the last time-step. We use the ℓ_2 norm to measure the similarity of the two hidden states. By minimizing Eq. 9, DLSTM is optimized to learn the most salient spatio-temporal patterns throughout the entire video that can characterize similar videos and distinguish videos of different labels. The above optimization can be performed by Back Propagation Through Time (BPTT) [3], which unfolds an LSTM model over several time steps and then runs the back propagation algorithm to train the model. To prevent back-propagated errors from decaying or exploding exponentially, we use truncated BPTT according to Hochreiter *et al.* [6] to learn the model parameters. Specifically, in our model, errors are not allowed to re-enter the memory cell once they leave it through the DoS nodes.

Fig. 3 shows the total training loss through the iteration process in the UCF101 data set [14]. We demonstrate the results of three different numbers of hidden states. As can be seen, the training loss decreases steadily through epochs and gradually converges when reaching 100 epochs. With a larger number of hidden states, the training process converges faster and achieves a lower converging training cost. The reason behind this is that with more hidden states, DLSTM has a more complex model and is more capable of fitting the training data.

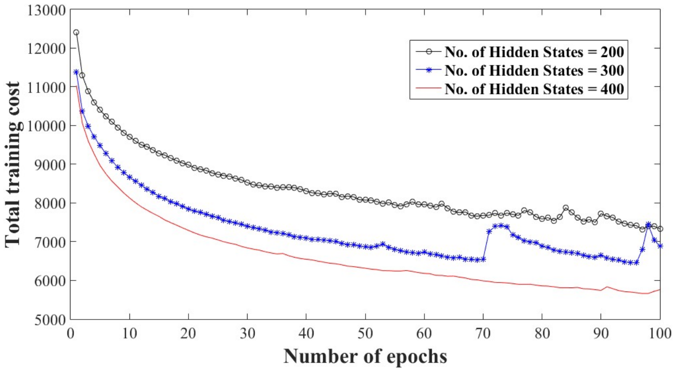

Fig. 3. training loss vs epochs on UCF101.

However, if the number of hidden states is too large, overfitting might occur.

C. Spatio-Temporal Feature Representation by DLSTM

Due to DLSTM's ability to capture long-term memory, the hidden states of the trained DLSTM at the last time-step have accumulated rich information throughout the entire video and thus provide a semantic representation of the whole video. Therefore, we use the hidden state at the last time-step to represent the entire video. We call this fixed-size feature representation the *DLSTM feature*, which provides a highly compact representation of the original video. In general, the number of hidden states will be moderate in order to avoid overfitting. Therefore, DLSTM-based modeling can provide a highly compact representation of the original video, which is suitable for large-scale video processing tasks. Different hashing algorithms can be directly applied to the DLSTM feature for video retrieval.

IV. Performance Evaluations

In this section, we extensively evaluate the performances of the proposed video modeling technique for hash-based video retrieval on two publicly available video data sets: UCF101 [14] and MSRActionPairs [11].

A. Experiment Setup and Performance Metrics

We perform the video retrieval experiments as follows. After the DLSTM is trained, the video will be fed to the DLSTM to produce the DLSTM feature, which will later be used by various hashing algorithms to generate hash bits. A query video is then used to retrieve similar videos in the database. Specifically, KNN is employed to search for the nearest neighbors by the Hamming distance on the hash codes. The retrieved videos are ranked by their similarity to the query video and the mean average precision (mAP) of the top 100 ranked results is used as the performance metric.

B. Experiments on the UCF101 data set

The UCF101 data set [14] is a large-scale video data set of human activities collected from YouTube. With 13,320 videos from 101 categories, UCF101 gives the largest diversity of classes among video data sets. Due to its large size and rich categories, UCF101 is a perfect candidate for evaluating the performance and efficiency of the large-scale video retrieval. In our experiments, we follow the "Three Train/Test Splits" settings in [14] and report the average results. To handle the huge number of video pairs in the training set, we randomly select 1% of the same-label pairs and a proportional number of the different-label pairs to train the DLSTM model. We apply the pretrained ILSVERC12 [12] model and the Caffe network [8] to each frame of the video and adopt the top layer output of the CNN as the original feature of video frames.

TABLE I
Impact of the number of DLSTM hidden states.

Num of hidden states	100	200	300	400	500
UCF101	38.52	43.13	45.40	**46.63**	42.33
MSRActionPairs	67.29	**74.39**	71.13	67.33	61.26

TABLE II
Comparison with different spatio-temporal modeling methods.

Modeling Method	DTW [10]	BoW [16]	DTQ [22]	DLSTM
UCF101	31.02	21.53	36.59	**46.63**
MSRActionPairs	–	–	62.37	**74.39**

The number of hidden states is an important factor in the performance of DLSTM-based video modeling. With an insufficient number of hidden states, the DLSTM cannot model the video effectively due to the underfitting effect. On the contrary, to avoid overfitting, the number of hidden states cannot be too large. To assess the impact of the number of hidden states, we perform the task of video retrieval by the Euclidean distance of the DLSTM feature. Experimental results are summarized in Table I. We use the aforementioned mAP as the performance metric. For the UCF101 data set, DLSTM achieves the highest mAP when the number of hidden states is 400. Since the dimension of the original feature vector of the video frames is 300, the optimal number of hidden states is slightly higher than the dimension of the input data. Similar results can be observed in the MSRActionPairs data set where the input feature dimension is 162 and the highest mAP is achieved with 200 hidden states.

Next, to assess DLSTM's performance on video modeling, we further compare the DLSTM-based modeling method with three other video modeling methods, namely Dynamic Time Warping (DTW), Bag-of-Words (BoW) and Dynamic Temporal Quantization (DTQ). These three methods are widely used for video classification/retrieval tasks and can also achieve a fixed-size representation of the video data. We implement the DTW-based motion template method in [10], the BoW method based on the HogHof feature detector [16] and the DTQ approach in [22] for comparison. In the experiment, we use the above three methods to generate a fixed-size feature representation of the video and perform the same video retrieval task according to the similarity by the Euclidean distance. Comparison results are shown in Table II. The DLSTM-based video modeling significantly outperforms all three methods in terms of mAP. This can be explained by the

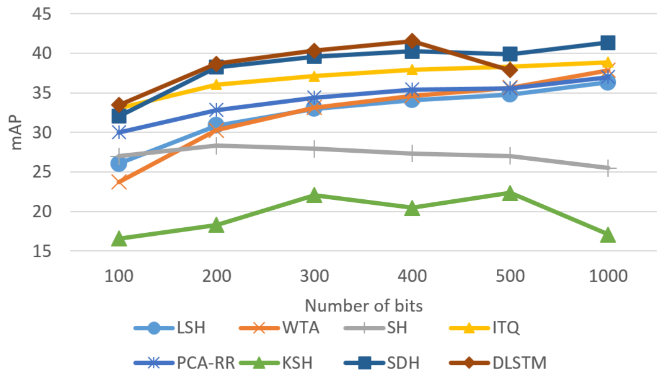

Fig. 4. Comparisons with baselines on UCF101.

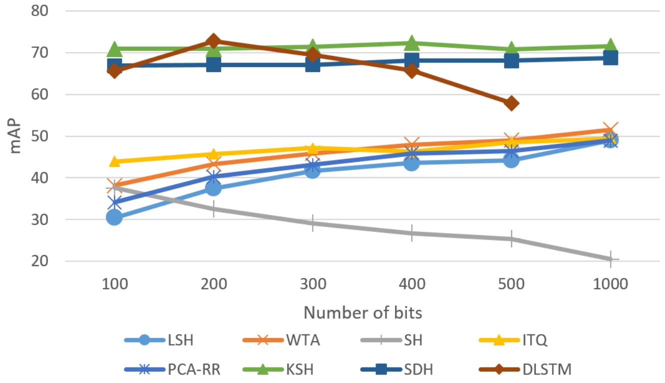

Fig. 6. Comparisons with baselines on MSRActionPairs.

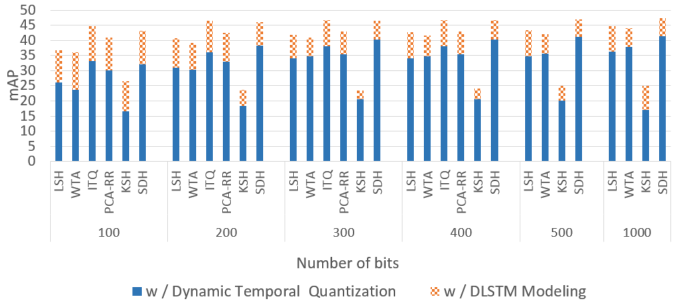

Fig. 5. Performance improvement of state-of-the-art hashing methods with DLSTM on UCF101.

fact that DTW, as a greedy sequence alignment method, may suffer from the misalignment of videos of varied lengths; while BoW only explores the local spatio-temporal features and does not leverage the global temporal information of the sequential data. Although DTQ achieves higher accuracy than DTW and BoW, it is still 10% lower than the proposed DLSTM-base modeling method. The above results demonstrate the superior performance of the DLSTM-based video modeling.

In the following part of the experiment, we evaluate the performance of the DLSTM-based modeling in the context of video hashing. The fixed-size DLSTM feature can be further encoded into hash bits with any existing hashing methods. For evaluation purposes, we adopt a very simple encoding method by quantizing the DLSTM feature vector into 0s and 1s according to the mean value of each dimension. Specifically, we first compute the mean value for each feature dimension. If a feature value is smaller or larger than the mean of corresponding dimension, it is encoded as 0 or 1, respectively. The reason to employ such a straightforward encoding scheme is to better assess DLSTM's contribution in the hash-based video retrieval. As discussed in the related work, most of the existing hashing techniques are for images. To achieve a reasonable comparison with state-of-the-art hashing algorithms, we create baseline methods that combine dynamic temporal quantization (DTQ) [22] and seven state-of-the-art hashing algorithms. Fig. 4 summarizes the results of the comparison. DLSTM with the simple encoding scheme has significantly outperformed most state-of-the-art hashing methods across different hash code lengths. The only method that performs close to DLSTM is Supervised Discrete Hashing (SDH). However, DLSTM achieves 41.57% mAP with only 400 bits while SDH needs much longer bits to achieve similar performance. Considering the large volume of the video data, DLSTM uses significantly fewer bits per frame for encoding and is therefore extremely efficient for large-scale video retrieval.

As mentioned in the previous section, DLSTM-based modeling can work with any existing hashing algorithm. We further apply state-of-the-art hashing algorithms on the DLSTM feature and compare the results with those of the aforementioned baselines. Experimental results are demonstrated in Fig. 5. It can be seen that all of the baselines have been further improved by $5\% - 15\%$ with the help of the DLSTM video modeling. The results demonstrate that the DLSTM-based video modeling can work with existing hashing methods to further enhance the video retrieval results.

C. *Results on the MSRActionPairs data set*

The MSRActionPairs data set [11] provides 3D videos captured by Kinect sensor. It provides a good variety of multimodal data streams, including RGB streams, depth streams and human skeleton joint streams. The database consists of 12 types of human actions performed by 10 subjects. Each subject repeats each action three times.

Although the MSRActionPairs data set has a relatively smaller size, it is a good candidate to evaluate the performance of the spatio-temporal modeling since the data set consists of human actions of similar postures but reverse temporal orders. Furthermore, the multimodal nature of the data set enables us to evaluate the proposed technique in multimodal data retrieval. We follow the same cross-subject test setting as in [11] and adopt the Histogram of Velocity Components (HVC) feature [22] to represent each frame of the 3D video.

Similar to the experiments on the UCF101 data set, we first compare the DLSTM with the baseline hashing methods. The results are shown in Fig. 6. Again, DLSTM, combined with the simple encoding scheme, has significantly outperformed most state-of-the-art hashing methods across different code lengths.

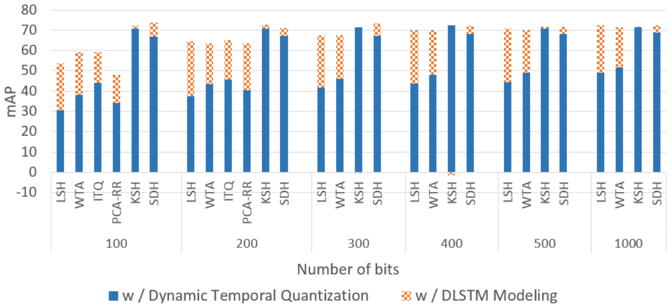

Fig. 7. Performance improvement of state-of-the-art hashing methods with DLSTM on MSRActionPairs.

Both KSH and SDH have close performances to DLSTM. However, DLSTM achieves higher performance with a shorter number of bits. Specifically, DLSTM achieves 72.72% mAP with only 200 bits, which is extremely efficient considering that the average number of frames in the MSRActionPairs data set is 112. In other words, it takes DLSTM less than two bits to encode a video frame on average. This shows DLSTM modeling is very suitable for large-scale video retrieval.

We also apply the DLSTM feature to state-of-the-art hashing methods and demonstrate the comparison results in Fig. 7. Similar to the results on the UCF101 data set, by using the DLSTM feature, most state-of-the-art hashing methods have increased to 70% mAP with 300 bits. The results again show that existing hashing methods significantly benefit from the DLSTM-based spatio-temporal modeling.

V. Conclusions

In this paper, we propose to study differential long short-term memory recurrent neural networks for modeling the spatio-temporal dynamics of videos. This approach can generate highly compact fixed-length representations for videos of varied lengths. The generated DLSTM feature can further benefit existing image hashing methods. Our extensive experimental results indicate that DLSTM modeling achieves competitive results even with a very simple hash function. When combined with state-of-the-art hashing techniques, DLSTM modeling substantially outperforms Dynamic Temporal Quantization.

VI. Future Work

In the current paper, we address the video modeling and video hashing via two steps and each step works independently. We would like to investigate the end-to-end DLSTM-based video hashing algorithm in the future by combining video modeling and video hashing.

VII. Acknowledgement

This material is based upon work partially supported by NASA under Grant Number NNX15AV40A. Any opinions, findings, and conclusion or recommendations expressed in this materials are those of the authors and do not necessarily reflect the views of NASA.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *Human Behavior Understanding*, pages 29–39. Springer, 2011.

[2] L. Cao, Z. Li, Y. Mu, and S.-F. Chang. Submodular video hashing: a unified framework towards video pooling and indexing. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 299–308. ACM, 2012.

[3] M. P. Cuéllar, M. Delgado, and M. Pegalajar. An application of non-linear programming to train recurrent neural networks in time series prediction problems. In *Enterprise Information Systems VII*, pages 95–102. Springer, 2007.

[4] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 817–824. IEEE, 2011.

[5] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[8] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[9] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2074–2081. IEEE, 2012.

[10] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the 2006 ACM SIGGRAPH*, pages 137–146.

[11] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[13] F. Shen, C. Shen, W. Liu, and H. Tao Shen. Supervised discrete hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 37–45, 2015.

[14] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[15] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[16] J. R. Uijlings, I. Duta, N. Rostamzadeh, and N. Sebe. Realtime video classification using dense hof/hog. In *Proceedings of International Conference on Multimedia Retrieval*, page 145. ACM, 2014.

[17] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. June 2015.

[18] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(12):2393–2406, 2012.

[19] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, pages 1290–1297, 2012.

[20] J. Yagnik, D. Strelow, D. A. Ross, and R.-s. Lin. The power of comparative reasoning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2431–2438. IEEE, 2011.

[21] G. Ye, D. Liu, J. Wang, and S.-F. Chang. Large-scale video hashing via structure learning. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2272–2279. IEEE, 2013.

[22] J. Ye, K. Li, G.-J. Qi, and K. A. Hua. Temporal order-preserving dynamic quantization for human action recognition from multimodal sensor streams. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 99–106. ACM, 2015.