REALISTIC HUMAN ACTION RECOGNITION: WHEN DEEP LEARNING MEETS VLAD

Lei Zhang[1], *Yangyang Feng*[1], *Jiqing Han*[2], *Xiantong Zhen*[3]

[1] College of Information and Communication Engineering, Harbin Engineering University, Harbin, PRC

[2] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, PRC

[3] The University of Western Ontario, London, ON, Canada

ABSTRACT

Human action recognition from realistic scenarios is extremely challenging due to large intra-class variation and complex background clutters. In this paper, by leveraging the strength of deep learning and vector of locally aggregated descriptors (VLAD), we propose a new methods for human action recognition from realistic datsets. We adopt stack convolutional independent subspace analysis (ISA) networks to learn 3D cuboid representation directly from spatio-temporal video data; we propose an improved VLAD by incorporating the spatio-temporal geometrical information to encode the deep learned local features.

On two challenging realistic datasets: the YouTube action and HMDB51 datasets, the proposed method achieves state-of-the-art performance with an efficient linear SVM classifier, which is competitive with and even better than existing sophisticated algorithms.

Index Terms— deep learning, convolutional ISA, VLAD, geometric information

1. INTRODUCTION

Recently, deep learning has achieved great success in many applications including video, image, speech and signal processing. Deep learning can be generally viewed as a model that extracts hierarchical information by stacking the basic model into different hierarchical structure. The basic model can be restricted Boltzmann machine (RBMs) [1, 2], independent component analysis (ICA) [3], independent subspace analysis (ISA) [4], sparse coding etc. RMBs have been successfully applied in deep belief network (DBN) [5] by treating each layer as an RBM and deep Boltzmann machine (DBM) [6] where the hidden units are organized in a deep layered manner and only adjacent layers are connected.

With respective to ICA, [3] shows that the filters learned by ICA on natural image data match very well with the classical receptive fields of cortical area V1 simple cells. Filters learned by sparse coding [7, 8] also give responses similar to cortical area V1 simple cells. In [9], sparsity and ICA are bond together for deep sparse filter. ISA and topographic ICA (TICA) [10] are two different extensions of ICA. ISA is a nonlinear version of ICA with much more robustness to local translation by properly selecting frequency, rotation and velocity. Moreover, TICA organizes features in a topographical map by pooling groups of related features, which is robust to local transformations.

On deep architectures, stacking basic model as in DBN [5] and DBM [6] with units and layers connections is widely used to build the hierarchical architectures for feature learning and classification. Recently, based on the fact that local statistic property can reflect the global one to a great extent, convolutional architecture which alternates convolutional layers and pooling layers becomes popular such as convolutional neural networks [11] for image classification. Stacked convolutional architecture is a prevalent choice to combine the advantages of convolutional structure for deep feature learning.

Human action recognition have extensively studied by learning spatio-temporal features [12, 13, 14, 15]. The bag of feature (BoF) model on local descriptors has been widely used for retrieval or classification tasks in image and human action recognition [16]. In contrast to BoF, the vector of locally aggregated descriptor (VLAD) [17] encode gradients or distances to all codewords to achieve soundable better performance than that of BoF.

Existing algorithms on VLAD is mainly based on hand-designed local descriptors. Recently, deeply hierarchically learned descriptor [18] has been successfully used to capture the structure information in a unsupervised way. However, to the best of our knowledge, deep learned local features have not been investigated in for the VLAD framework. To fill this gap, we in this paper propose leveraging the respective strengths of convolutional ISA and VLAD in unsupervised feature learning and local feature encoding for human activity recognition.

2. DEEP LEARNED SPATIO-TEMPORAL FEATURE

We propose adopting the stacked convolutional independent subspace analysis (ISA) to leverage its great effectiveness in unsupervised feature learning.

This work is partially sponsored by National Science Foundation of China (No. 61571147 and No. 91220301), National Science Foundation of Heilongjiang (F2015027)

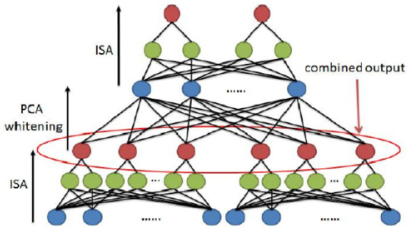

Fig. 1. The illustration of the stacked convolutional ISA network [18].

2.1. Stacked convolutional ISA

The basic unit in stacked ISA can be viewed as a two-layered network and the whole architecture in Fig. 1 contains two layers of ISA. The first layer is composed of multiple ISAs with the input of cuboids densely extracted from a video sequence. Treating ISA as a filter or a feature learning kernel, it covers all possible cuboids in a video. The combined outputs of each ISA in the first layer after PCA dimension reduction and whitening, are fed into the second layer of ISA.

For the finally learned features, we combine features from the outputs of the first followed by a PCA whitening and second layers, which has shown satisfactory performance while maintaining computational efficiency [18].

Despite cuboid representation deep learned by stacked convolutional ISA, geometric information are further combined as two additional dimensions to compensate the location information loss in VLAD.

2.2. Geometric information

In conventional BoF, VLAD or FV frameworks, the important information about spatial distribution is lost due to the feature pooling strategy. In image representations, spatial pyramid matching (SPM) [19, 20] holds dominant position in spatial compensation for its remarkable success in image classification tasks. The key idea of SPM involves repeatedly subdividing the image and computing histogram of local features at increasingly fine resolutions. The main shortcoming of SPM lies in the increasing dimension in BoF, which is 21 times of the original codebook size for a three-level pyramid, which has 1, 4, 16 grids for an image.

[21] provides a different way to add geometric information by a global weight related to locations. It aims to enhance the discriminative ability since the weight is learned by marginal fisher analysis. The main problem for this approach is the global weight learning procedure, which fuses too much information together as location information, intra-class similarity information and inter-class discriminative information. In this hybrid weight, geometric information is weakened.

However, [21] shows that the local descriptors indexed by the same codeword often share similar spatial distributions.

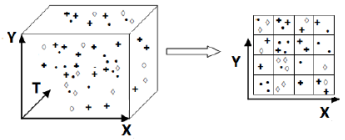

Fig. 2. Toy demonstration of spatial projection in one video, where the temporal information is neglected

In fact, real data always exhibits some smoothness properties and locality properties. It means that the similar descriptors always share close locations. Inspired by the locality property, we propose a simply but very effective way to capture the geometric information to compensate the structural loss problem in VLAD.

If we neglect the temporal information and project the 3D cube into x-y plane, we can see that the similar descriptors located into the same grids as shown in Fig. 2. We propose to concatenate the normalized grid location information (x, y) which is in the range of $[0, 1]$ at the end of hierarchical learned spatio-temporal features.

3. VLAD ON DEEP LEARNED SPATIO-TEMPORAL FEATURE

3.1. Revisit of VLAD

Given a visual codebook $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K\}$ which is off-line learned by the k-means algorithm, then the construction of VLAD can be decomposed into four steps:

1) Local descriptor: Normally this local invariant descriptor is as SIFT or HOG/HOF. When combined with hierarchically learned local descriptors, it is the output of Fig. 1. We re-denote the output as $\mathbf{x}^i$.

2) Residual vector: It is the subtraction between the local invariant descriptor and its belonging codeword $\hat{c}$, which is as $\mathbf{x}^i - \mathbf{u}_{\hat{c}}$, where $\hat{c}$ is obtained by $\hat{c} = \arg\min_c |\mathbf{x}^i - \mathbf{u}_c|$.

3) Pooled vector: In standard VLAD, it is sum pooling in this level. Pooled vector only depends on the local descriptors assigned into the same codeword. The sum pooling is as:

$$\mathbf{v}_{\hat{c}} = \sum_{\mathbf{x}^i \in \hat{c}} (\mathbf{x}^i - \mathbf{u}_{\hat{c}}) \tag{1}$$

4) Aggregated vector: Concatenate K pooled vectors as a single VLAD $[\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K]$, where the dimension of VLAD is $K \times d$.

3.2. Normalization in VLAD

Generally, there are two types of normalization strategies, i.e., ℓ_2 normalization and power-law normalization. Since the former is conducted on the whole vector while the latter is a component-wise operation, these two normalization strategies

can be further combined. If different normalization strategies are conducted on distinct vectors, there are many combinations with different performance.

Based on the above experiments on different normalization combinations of VLAD on SIFT, we select one of the top combinations here with only ℓ_2 normalization on local descriptors, pooled vectors and aggregated vectors.

3.2.1. Normalization on deep learned descriptors

Applying ℓ_2 normalization on deep learned descriptors will effect the performance from following aspects:

Codebook generation: The obtained codebook generated on normalized local descriptors will be significantly different from that without normalization. No matter what kind of local descriptor is adopted, each sample is just a point in high-dimensional space. With the increase of dimensionality, the 'curse of dimension' will be much more obvious and the data is of much more sparse.

Before ℓ_2 normalization, the generated codewords can be scattered at any positions in this high dimensional space, which leads to large uncertainty and ambiguity among codewords. However, after ℓ_2 normalization, all samples are relocated on the surface of a hyper-sphere, so the sparsity problem can be alleviated to a large extent. Under this condition, the codeword which is the cluster center of some local descriptors will fall within the unit hyper-sphere and nearby the surface.

Assignment strategy: When ℓ_2 normalization is adopted, the Euclidean distance provides a similar measurement of distances to the cosine distance since the influence of the vector magnitude is eliminated, which leads to a total different assignments $\hat{c}$ in Eq.(1) and produces a big difference for the pooled vector.

3.2.2. Normalization on pooled vectors

With respect to the normalization on pooled vectors, we analysis the effects from following two aspects:

Burstiness alleviation: Burstness firstly noticed in BoF [22] is the property that a given visual element appears much more times than a statistically independent model could predict. In [22], burstiness of visual words is shown to affect the performance. Both [23] and [17] point out that in VLAD, the phenomenon does exist that the dimensions with high values in VLAD are mostly from the same clusters.

ℓ_2 normalization on pooled vectors in [23] is called intra-normalization, which is used to force the burst cluster with the similar magnitude with those from other clusters. After this normalization, all pooled vectors from different codewords have the similar magnitude.

Codeword Independence: The essence of aggregated vectors is to treat pooled vectors from different codeword independently. In order to eliminate the negative effect of different magnitudes of pooled vectors, ℓ_2 normalization is a soundable selection to keep intrinsic property of the pooled vector unchanged before aggregation.

layer	Accuracy	MAP	codebook size
The first ISA layer	81.05%	80.93%	256
The second ISA layer	80.48%	79.59%	64
The first ISA + the second ISA layers	82.10%	81.54%	64

Table 1. Performance with different layer output with optimized codebook size on YouTube action dataset

3.2.3. Normalization on aggregated vectors

The ℓ_2 normalization on the whole VLAD representation will further eliminate the magnitude influence among different videos from distinct categories.

4. EXPERIMENTS AND DISCUSSIONS

The cuboid size from 3D video is $20 \times 20 \times 14$, which is the same as [18]. Since the performance of dense sampling is better than sparse sampling both for image classification and video processing [18, 24], the cuboid is densely sampled over the whole video. Two datasets are selected as follows:

The Youtube action dataset [25]: This is a challenging dataset due to large variations in camera motion, object appearance and pose, viewpoint and so on. This dataset contains 1168 sequences from 11 action categories.

HMDB51 dataset [26]: The HMDB51 dataset is a large dataset containing 6766 video clips extracted from various sources, ranging from movies to YouTube. It consists of 51 action classes, each having at least 101 samples and some samples are with dark and cluttered background.

4.1. Performance on the YouTube action dataset

We conduct extensive experiments on the YouTube dataset to comprehensively investigate the proposed method and determine the optimal parameter settings.

Deep learning benefit: In this part we compare the performance of the first ISA layer and the second ISA layer separately and jointly. Table 1 reports the best performance for both accuracy and MAP with optimized codebook sizes. From the system performance in Table 1, it has been demonstrated that the effect of the first ISA layer and the second ISA layer should not be removed or replaced each other. The information provided by different layers are complementary and the performance with both the first ISA layer and the second ISA layer can achieve better result compared with performance of a single layer.

Geometric information benefit: To compensate the loss of geometric information in the VLAD framework, x and y

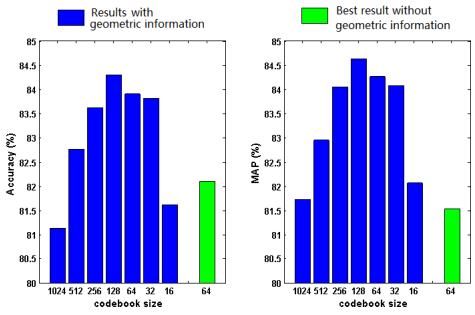

Fig. 3. Performance of VLAD with hierarchically learned descriptors with geometric information for YouTube action dataset

Algorithm	Accuracy
Hierarchical feature on ISA + BoF [18]	28.10%
Hierarchical feature on ISA + VLAD + geometric information (with 256 codewords)	**33.10**%
Sparse + HOG + HOF +BoF [26]	21.96%
Sparse + C2 +BoF [26]	23.18%
Dense cuboids + HOG + BoF [27]	25.20%
Dense cuboids + HOF + BoF [27]	29.40%
Dense cuboids + MBH + BoF [27]	40.90%
Dense cuboids + HOG + HOF + MBH +BoF [27]	43.10%
Action Bank [28]	26.90%

Table 2. Performance comparison with state-of-the-art approaches on the HMDB51 dataset

spatial location information are further added after normalization. Fig. 3 shows the performance of VLAD on hierarchically learned descriptors with geometric information on the YouTube action dataset. It is clear to see the performance enhancement with the geometric information compensation, which can achieve 84.29% for accuracy and 84.62% for MAP with the codebook size of 128. In addition, it is worth to note that even when codebook size is only 32, the accuracy and MAP are over 83.5%.

4.2. Performance on the HMDB51 dataset

Since the benefit of the hierarchical architecture on the YouTube action dataset is experimentally determined, on the HMDB51 dataset, we conduct the experiments on the combination of descriptors from the first ISA layer and the second ISA layer. Moreover, geometric information is added as default.

The top block in Table 2 gives the performance of the proposed method compared with BoF. For hierarchically learned descriptors, adopting VLAD rather than BoF can achieve about 5% improvement from 28.10% to 33.10%.

Algorithm	Accuracy
Hierarchical feature on ISA + BoF + Chi-square kernel [18]	75.80%
Hierarchical feature on ISA + BoF + Linear kernel	74.86%
Hierarchical feature on ISA + VLAD + geometric information	**84.29**%
Dense cuboids + HOG + BoF [27]	77.00%
Dense cuboids + HOF + BoF [27]	68.30%
Dense cuboids + MBH + BoF [27]	78.40%
Dense cuboids + HOG + HOF + MBH +BoF [27]	81.40%
Static + motion feature [25]	71.20%
Relative motion descriptor (RMD) + Modes [29]	81.70%
Dense trajectory + BoF [30]	84.20%

Table 3. Performance comparison of accuracy with state-of-the-art approaches on YouTube action dataset

4.3. Comparison to state-of-the-art performance

Table 2 and 3 summarize the performance of our proposed method and compare to state-of-the-art performance on the HMDB51 and YouTube action datasets.

On the YouTube action dataset, compared with the hierarchical feature from ISA with the BoF framework, no matter what kind of SVM kernel is adopted, the performance of hierarchical features from ISA combined with VLAD is much better. Moreover, jointly considering geometric information, the performance has been further improved, which is among the top level performance compared with existing algorithms in Table 3.

The performance of the proposed method on the HMDB51 dataset is competitive with existing sophisticated algorithms as shown in Table 2. The improvement of VLAD on hierarchically learned descriptors can also be found on the HMDB51 dataset in Table 2. Since the enhancement partly depends on the representation ability of local descriptors, it also reflects the fact that the unsupervised learning procedure in our hierarchical architecture can learn discriminative representations, as for the data from videos of complex background and large categories condition on the HMDB51 dataset.

5. CONCLUSION

In this paper, we have presented a new method to combine deep learned descriptor with VLAD. Experiments have been carried out on two challenging datasets: YouTube action and HMDB51. The proposed method by leveraging the strength of deep learning and VLAD has produces competitive performance with state-of-the-art algorithms. We can draw some conclusions from the results. Firstly, VLAD boosts the descriptors and exhibits a significantly better performance. Secondly, combining multi-layer representation in hierarchical learned descriptors is preferable for better performance. Compared with hand-craft local descriptors including HOG and HOF, hierarchical learned descriptor performs much better.

6. REFERENCES

[1] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507.

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *NIPS*, 2007.

[3] J. Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings: Biological Sciences*, vol. 265, no. 1394, pp. 359–366.

[4] A. Hyvarinen, J. Hurri, and P. Hoyer, "Natural image statistics," 2009.

[5] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554.

[6] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Proceeding of the international conference on artificial intelligence and statistics*, 2009.

[7] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609.

[8] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *NIPS*, 2007.

[9] N. Jiquan, K. Wei, Z. Chen, S. Bhaskar, and A. Ng, "Sparse filterin," in *NIPS*, 2011.

[10] Q. Le, J. Ngiam, Z. Chen, D. Chia, P. Koh, and A. Ng, "Tiled convolutional neural networks," in *NIPS*, 2010.

[11] A Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] X. Zhen, L. Shao, and X. Li, "Action recognition by spatio-temporal oriented energies," *Information Sciences*, 2014.

[13] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal laplacian pyramid coding for action recognition," *IEEE Transactions on Cybernetics*, 2013.

[14] X. Zhen, L. Shao, D. Tao, and X. Li, "Embedding motion and structure features for action recognition," *IEEE TCSVT*.

[15] X. Zhen and L. Shao, "Introduction to human action recognition," *Wiley Encyclopedia of Electrical and Electronics Engineering*.

[16] X. Zhen and L. Shao, "A local descriptor based on laplacian pyramid coding for action recognition," *Pattern Recognition Letters*, 2012.

[17] H. Jegou, F. Perronnin, M. Douze, and J. Sanchez, "Aggregating local image descriptors into compact codes," *IEEE T-PAMI*, 2012.

[18] Quoc V. Le, Zou. Y, S. Yeung, and A. Ng, "Learning hierachical invariant spatio-temporal feature for action recognition with independent subspace analysis," in *CVPR*, 2011.

[19] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[20] M Sekma, M. Mejdoub, and C. Amar, "Spatio-temporal pyramidal accordion representation for human action recognition," in *ICASSP*, 2014.

[21] J. Feng, B. Ni, Q. Tian, and S. Yan, "Geometric lp-norm feature pooling for image classification," in *CVPR*, 2011.

[22] H. Jegou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *CVPR*, 2009.

[23] R. Arandjelovic and A. Zisserman, "All about vlad," in *CVPR*, 2013.

[24] J. Sachez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *IJCV*, vol. 105, no. 3, pp. 222–245.

[25] J. Liu, J. Luo, and M. Shah, "Recogning realistic actions from videos "in the wild"," in *CVPR*, 2009.

[26] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *ICCV*, 2011.

[27] H. Wang, A. Klaser, and C. Schmid, "Dense trajectories and motion boundary descriptors for action recognition," *IJCV*, 2013.

[28] S. Sadanand and H. Corso, "Action bank: a high-level representation of activity in video," in *CVPR*, 2012.

[29] O. Oshin, A. Gilbert, and R. Bowden, "Capturing relative motion and finding modes for action recognition in the wild," *Int. Journal Computer Vision and Image Understanding, CVIU*, vol. 125, pp. 155–171, 2014.

[30] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011.