CODEBOOK ENHANCEMENT OF VLAD REPRESENTATION FOR VISUAL RECOGNITION

Zhe Wang[1,2] *Yali Wang*[1] *Limin Wang*[1,2] *Yu Qiao*[1]

[1]Shenzhen key lab of Comp. Vis. & Pat. Rec., Shenzhen Institutes of Advanced Technology, CAS, China
[2]Department of Information Engineering, The Chinese University of Hong Kong
{buptwangzhe2012,07wanglimin}@gmail.com, {yl.wang,yu.qiao}@siat.ac.cn

ABSTRACT

Recent studies demonstrate the effectiveness of super vector representation in a number of visual recognition tasks. One popular approach along this line is the Vector of Locally Aggregated Descriptor (VLAD) where the super vector is encoded with a codebook generated by *k*-means. However, the effectiveness of the codebook is often limited, due to the poor clustering solution, the high dimensionality of visual descriptors and the global PCA for data preprocessing. To circumvent these problems, we propose three approaches for codebook enhancement, (i) partition of data, (ii) partition of feature, and (iii) local PCA. Moreover, all these approaches can be effectively integrated together to further boost the recognition performance. In our experiments, we evaluate our enhancement approaches on two challenging visual tasks, i.e., action recognition (HMDB51) and object recognition (PASCAL VOC2007). The results show that our approaches and the fusion versions significantly outperform the baselines.

Index Terms— Visual Recognition, VLAD, *K*-Means, P-CA

1. INTRODUCTION

Visual recognition has been an important problem in many research areas, such as multimedia, computer vision, machine learning [1, 2, 3, 4]. Learning an effective visual representation is challenging due to the large intra-class variations in the real-world videos [5, 6, 7] and images [8, 9]. One popular feature encoding approach is the Vector of Locally Aggregated Descriptor (VLAD) which has been successfully used for several visual recognition tasks [2, 10].

A standard VLAD pipeline is as follows. *First*, a training feature set is extracted from T training videos (or images), and then processed as $\{\mathbf{X}^t\}_{t=1}^T$ by using the standard PCA (In the following, it is called as the global PCA for clarification). For the t-th video (or image), there are N_t feature vectors, i.e., $\mathbf{X}^t = \{\mathbf{x}_n^t\}_{n=1}^{N_t}$ and $\mathbf{x}_n^t \in \mathbb{R}^D$. *Second*, N feature vectors are sampled from $\{\mathbf{X}^t\}_{t=1}^T$ and used in k-means clustering to learn a codebook $\{\mu_j\}_{j=1}^k$, where μ_j is the mean of the j-th cluster. *Third*, a super vector for the t-th video (or image) is concatenated by $\mathbf{v}^t = [\mathbf{v}_1^t, ..., \mathbf{v}_k^t] \in \mathbb{R}^{k \times D}$, where $\mathbf{v}_j^t$ is yielded by aggregating the difference between $\mathbf{x}_n^t \in \mathbf{X}^t$ and μ_j. Specifically, $\mathbf{v}_j^t = \sum_{\mathbf{x}_n^t : NN(\mathbf{x}_n^t)=j} (\mathbf{x}_n^t - \mu_j)$ where $NN(\mathbf{x}_n^t) = j$ denotes that the nearest neighbour of $\mathbf{x}_n^t$ in the codebook is μ_j. *Finally*, the super vectors $\{\mathbf{v}_j^t\}_{t=1}^T$ are normalized and used for visual recognition.

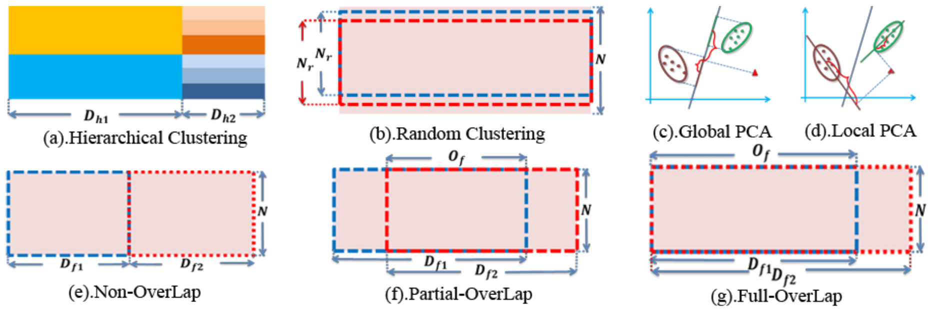
From the description above, we can see that codebook plays a key role to encode visual features in VLAD. However, the effectiveness of k-means is often limited, due to the following reasons. *First*, k-means depends on its initialization. This may result in a poor clustering solution which is not sufficient to explore the complex structure of visual data sets. *Second*, the high dimensionality of visual descriptors often degenerates the resulting clusters of *k*-means, which leads to a poor codebook. *Third*, the global PCA is often used in VLAD for data preprocessing. However, this may be not optimal to explore the local manifold in the feature space.

To address these problems, we propose the following methods for codebook enhancement in VLAD. **(i) Partition of Data**. We design two approaches to partition the training data so that k-means is performed with different partition manners to alleviate poor clustering solution. **(ii) Partition of Feature**. We introduce three feature-dimension partition approaches to reduce the influence of high-dimensionality of visual descriptors and thus better exploit the interconnection between different feature dimensions. **(iii) Local PCA**. We explore a local PCA approach to investigate the local manifold structure of different clusters. **(iv) Fusion of Different Methods**. We incorporate these methods into several fusion versions which take advantages of different approaches complementarily to generate more powerful codebooks. Our proposed enhancement approaches are simple but quite effective, and the experimental results on the real-world videos and images show that our approaches and the fusion versions can significantly boost the recognition performance.

2. RELATED WORK

Many research efforts have been devoted to improving VLAD representation. Delhumeau et al [11] proposed local coordinate system to handle "visual burstiness". Peng et al [12] introduced supervision in generating codebook. In our work, we focus on solving the poor clustering solution which is not sufficient to explore the complex structure of visual data sets and verify our methods on visual classification task.

978-1-4799-9988-0/16/$31.00 ©2016 IEEE



(a).Hierarchical Clustering (b).Random Clustering (c).Global PCA (d).Local PCA

(e).Non-OverLap (f).Partial-OverLap (g).Full-OverLap

Fig. 1. Illustration of Our Codebook Generation Methods

3. CODEBOOK ENHANCEMENT

In this section, we describe the proposed methods for codebook enhancement in VLAD. For simplicity, we organize the sampled N feature vectors (for codebook generation) into a $N \times D$ matrix, where the dimensionality of each feature vector is D. Note that this matrix is the one after PCA preprocessing, where D is the dimensionality that preserves 90% energy of the original feature. Additionally, the columns of this $N \times D$ matrix are in the descending order of energy.

3.1. Partition of Data

We propose two partition-of-data methods in which k-means is performed with different data partition mechanisms to alleviate poor clustering solution.

(i) Hierarchical Clustering (HC). We design an energy-based hierarchical structure to cluster the codebook matrix. First, we perform k-means on the first D_{h1} dimensions of the feature vectors to generate S_{h1} clusters. For each of S_{h1} cluster, we retrieve the indices of the original D dimensional vectors. By doing so, we obtain S_{h1} clusters for the D dimensional vectors, based on the D_{h1} most energetic dimensions. Then, for each of these S_{h1} clusters, we perform k-means on the rest $(D - D_{h1})$ dimensions of the feature vectors to generate $k = S_{h2}$ clusters. Consequentially, we divided S_{h1} clusters into $S_{h1} \times S_{h2}$ clusters for the D dimensional vectors, based on the remaining $(D - D_{h1})$ energetic dimensions. Compared to one-time k-means on the entire matrix, **HC** performs k-means multiple times via a $L_h = 2$ layer structure. It allows us to cluster the D dimensional feature vectors from coarse to fine, thus produce a reasonable codebook. The illustration of HC is shown in Figure 1(a).

(ii) Random Clustering (RC). The second partition approach is based on random sampling within the data matrix. Specifically, we randomly sample data T_r times from the codebook matrix. For each of T_r times, N_r ($N_r < N$) feature vectors are sampled and then used in k-means to generate S_r clusters. Hence, this RC method totally produces $T_r \times S_r$ clusters. Compared to one-time k-means, **RC** randomly performs k-means multiple times to avoid poor clustering solution. The illustration of RC is shown in Figure 1(b).

3.2. Partition of Feature

In traditional codebook generation, k-means is performed on the entire D dimensional feature vectors. However, the effectiveness of k-means is limited for high-dimensional vectors, due to the curse of dimensionality. Inspired by product quantization [13], we propose to partition the D dimensional vector into a number of low-dimensional vectors to boost the performance of k-means and enhance the codebook generation. **(i) Non-OverLap (N-OL)**. We divide the $N \times D$ codebook matrix into a $N \times D_{f1}$ matrix and a $N \times D_{f2}$ matrix where $D = D_{f1} + D_{f2}$. **(ii) Partial-OverLap (P-OL)**. To further exploit the correlation of different feature dimensions, we propose to share O_f dimensions between the $N \times D_{f1}$ and $N \times D_{f2}$ matrix. In this case, $D = D_{f1} + D_{f2} - O_f$. **(iii) Full-OverLap (F-OL)**. We consider a full-overlap case where $O_f = D_{f1}$, $D = D_{f2}$. For all these 3 cases, k-means is respectively performed on the $N \times D_{f1}$ and $N \times D_{f2}$ matrices to generate S_f clusters for each matrix. The illustrations of N-OL, P-OL and F-OL are shown in Figure 1(e-g).

Table 3. Local PCA (HMDB51)

Method	HOG	HOF	MBHx	MBHy	Total
VLAD$_{256}$	35.90	46.70	38.40	43.20	55.50
LCS[11]	32.53	41.90	34.10	40.26	53.07
LPCA	36.25	46.69	37.71	44.42	**59.00**

3.3. Local PCA

Traditionally, global PCA is adopted to preprocess training descriptors before codebook generation. This leads to the fact that all clusters generated by k-means share the same dimension reduction matrix. Hence, it may ignore the local data structure in each cluster and reduce the power of codebook. For instance, in Figure 1(c), training sample A is actually closer to cluster C_2 in original 2 dimensional space. However, a single PCA projects all 2-D training data to a line. On this line, the projected point of center of cluster C_1 is closer to the projected point of A. Thus, VLAD mistakenly choose the distance between two projected points to encode A.

We propose a local PCA (LPCA) approach. After generating the clusters by k-means, we retrieve the indices of feature vectors in each cluster to find the original training vectors (data before PCA preprocessing). Then, we perform PCA for each cluster by using all the original training vectors in that cluster. Thus all the clusters have their own dimensionality-reduction matrix to explore the local manifold structure of codebook. Illustration of LPCA is shown in Figure 1(d). LP-

Table 1. Partition of Data (HMDB51)

Partition of Data: HC	L_h	D_{h1}	D_{h2}	S_{h1}	S_{h2}	Total
VLAD$_{256}$	1	48	N/A	256	N/A	55.50
HC-VLAD	2	36	12	128	2	**57.63**

Partition of Data: RC	$N_r \times T_r \times S_r$	HOG	HOF	MBHx	MBHy	Total
VLAD$_{256}$	100w×1×256	35.90	46.70	38.40	43.20	55.50
RC-VLAD	70w×2×128	35.38	47.41	38.47	44.79	**58.61**

Table 2. Partition of Feature (HMDB51)

Partition of Feature	D_{f1}	D_{f2}	O_f	S_f	HOG	HOF	MBHx	MBHy	Total
VLAD$_{256}$	48	N/A	N/A	256	35.90	46.70	38.40	43.20	55.50
N-OL-VLAD	24	24	N/A	256	37.65	47.71	37.28	43.40	**57.34**
VLAD$_{512}$	48	N/A	N/A	512	36.93	47.70	39.43	44.27	58.26
P-OL-VLAD	36	36	24	256	39.04	48.95	39.46	45.62	58.54
F-OL-VLAD	36	48	36	256	38.21	48.37	39.83	45.49	**59.30**

Table 5. Comparison with Related Methods

Method	HMDB51	Method	VOC07
Wang et al. [14]	42.1	Vedaldi et al. [15]	54.7
Jain et al. [16]	52.1	Russakovsky [17]	57.2
Wang et al. [3]	57.2	Chatfield [18]	61.7
Ours	**59.8**	Ours	**63.2**

CA is different from local coordinate system(LCS) [11]. LCS first clusters descriptors without dimension reduction using k-means, and for each cluster they learn a rotation matrix. Dimension reduction can be based on this rotation matrix.

3.4. Fusion with Different Methods

Finally, we integrate the above three methods to further improve the recognition performance. For instance, the partition-of-data methods can be used to generate clusters for the partition-of-feature methods while Local PCA can be used to obtain a distinct dimensionality-reduction matrix for each cluster generated by the partition-of-data methods.

4. EXPERIMENTS

We verify the effectiveness of our methods on a human action recognition data (HMDB51 [19]) and an image classification data (PASCAL VOC2007 [20]). HMDB51 consists of 51 action categories with 6,766 annotated videos. We use low-level descriptors of HOG, HOF, MBHx, MBHy for video representation. We perform experiments on three training/testing splits which are released on the official website, and report average predictive accuracy for evaluation. PASCAL VOC2007 consists of 20 different object categories with 5,011 training and 4,952 test images. The low-level feature used in our approach is SIFT. We report the mean of AP (mAP) over 20 categories using the standard PASCAL protocol.

For HMDB51, the baseline for comparison is VLAD$_{256}$ [12] where the codebook has 256 clusters, and the dimensionality of HOG/HOF/MBHx/MBHy (after global PCA) is 48/54/48/48. Except mentioned, we choose the following setting for HOG/HOF/MBHx/MBHy in order to make a fair comparison with VLAD$_{256}$. The setting of **HC** is (L_h, D_{h1}, D_{h2}, S_{h1}, S_{h2}) = (2, 36/40/36/36, 12/14/12/12, 128, 2); **RC** is (N_r, T_r, S_r) = (70w, 2, 256); **N-OL** is (D_{f1}, D_{f2}, O_f, S_f) = (24/27/24/24, 24/27/24/24, 0, 256); **P-OL** is (D_{f1}, D_{f2}, O_f, S_f) = (36/40/36/36, 36/40/36/36, 24/26/24/24, 256); **F-OL** is (D_{f1}, D_{f2}, O_f, S_f) = (36/40/36/36, 48/54/48/48, 36/40/36/36, 256). Next we will use the setting for HOG to explain how to perform our experiment. The analysis for HOF, MBHx, MBHy is similar.

Partition of Data (PoD): To compare to VLAD$_{256}$, we design **(i) HC** with a $L_h = 2$ layer hierarchical structure to generate $S_{h1} \times S_{h2} = 128 \times 2 = 256$ clusters, and **(ii) RC** which is performed with $T_r \times S_r = 2 \times 128 = 256$ clusters. We denote these two approaches as HC-VLAD and RC-VLAD. Table 1 shows that both HC-VLAD and RC-VLAD outperform VLAD$_{256}$. It illustrates that both HC and RC generate a more informative codebook by performing k-means through a energy-based hierarchical structure (in HC) or random sampling (in RC). **(iii) HC vs RC**. Note that RC-VLAD tends to outperform HC-VLAD in Table 1. This is ascribed to data overlap and random initialization offset when using RC.

Partition of Feature (PoF): **(i) N-OL**. We divide the $D = 48$ dimensional feature space into two non-overlap feature subspaces ($D_{f1} + D_{f2} = 24 + 24 = 48$). For each feature subspace, there are $S_f = 256$ clusters. Thus, our codebook is $(D_{f1} + D_{f2}) \times S_f = 48 \times 256$ which is the same as VLAD$_{256}$. We denote it as N-OL-VLAD. Table 2 shows N-OL-VLAD outperforms VLAD$_{256}$. The main reason is that N-OL divides the high-dimensional space into a number of low-dimensional subspaces. K-means clustering is generally more effective in low-dimensional subspace, and generates a more informative codebook. **(ii) P-OL & F-OL**. To further explore the interconnection between different feature dimensions, we divide the $D = 48$ dimensional feature space into two overlapped feature subspaces. For P-OL, $D_{f1}+D_{f2}-O_f = 36+36-24 = 48$. We denote it as P-OL-

Table 4. Fusion with Different Methods (HMDB51)

Method	PoD		PoF	LPCA	Accuracy				
	HC	RC	F-OL		HOG	HOF	MBHx	MBHy	Total
VLAD$_{256}$	x	x	x	x	35.90	46.70	38.40	43.20	55.50
VLAD$_{512}$	x	x	x	x	36.93	47.70	39.43	44.27	58.26
RC+F-OL	x	✓	✓	x	39.06	49.96	41.00	46.69	59.00
RC+LPCA	x	✓	x	✓	38.54	49.28	40.20	46.38	59.26
F-OL+LPCA	x	x	✓	✓	38.89	49.04	42.72	46.12	59.32
RC+F-OL+LPCA	x	✓	✓	✓	39.52	49.06	41.11	47.25	59.54
HC/RC+F-OL+LPCA	✓	✓	✓	✓	**40.37**	**50.46**	**42.83**	**48.74**	**59.79**

VLAD. For F-OL, $D_{f1} + D_{f2} - O_f = 36 + 48 - 36 = 48$. We denote it as F-OL-VLAD. Additionally, for each feature subspace in both approaches, $S_f = 256$. Hence, the dimensions of P-OL-VLAD and F-OL-VLAD are respectively $(D_{f1}+D_{f2})\times S_f = 72\times 256 = 36\times 512$ and $(D_{f1}+D_{f2})\times S_f = 84 \times 256 = 42 \times 512$. To be fair, we switch baseline from VLAD$_{256}$ to VLAD$_{512}$[21] in which the codebook is 48×512. Table 2 shows that both P-OL-VLAD and F-OL-VLAD outperform VLAD$_{512}$, even the dimensions of our two approaches are lower than VLAD$_{512}$. This indicates that P-OL and F-OL help VLAD construct a more informative codebook with the overlapped low-dimensional subspaces. **(iii) N-OL vs OL**. P-OL-VLAD & F-OL-VLAD tend to outperform N-OL-VLAD in Table 2, and F-OL-VLAD perform the best. The main reason is that P-OL & F-OL capture the interconnection between feature dimensions, and F-OL achieves the most powerful codebook with a fully-overlapped subspace.

Local PCA: Besides VLAD$_{256}$, we implement LCS [11] for comparison. In Table 3, LPCA outperforms both VLAD$_{256}$ and LCS[11]. This is because LPCA captures local manifold structure of each cluster and generate more informative codebooks.

Fusion with Different Enhancement Mechanisms. Results of our fusion versions are in Table 4. Since there are multiple methods in PoD and PoF, we choose the method with the best performance in previous experiments. Hence, for PoD we choose RC; for PoF we choose F-OL. As expected, all the two-method fusion approaches (RC+F-OL, RC+LPCA, F-OL+LPCA) outperform the two baselines. Moreover, F-OL+LPCA achieves a better result than RC+F-OL and RC+LPCA. This indicates that F-OL and LPCA are more effective to generate a powerful codebook. Then we combine all these three methods as RC+F-OL+LPCA. This fusion version further improves the performance of two-methods fusion approaches.We notice that HC can be incorporated into RC. Hence we explore to incorporate this fusion into RC+F-OL+LPCA. HC/RC+F-OL+LPCA approach significantly outperforms the two baselines in table 4.

Properties of Our Approaches:. We explore the robustness of our approaches to the parameter settings. Since RC performs the best among PoD approaches and F-OL performs best among PoF approaches, we here show the accuracy of RC and F-OL as a function of different parameter settings. In Figure 2, both RC and F-OL are robust to different parameters for different features.

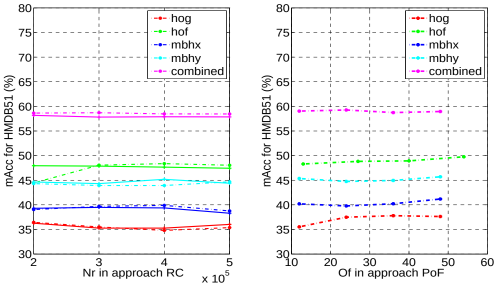


Fig. 2. Accuracy as a function of different parameter settings Nr in our approaches PoD(dotted line:T_r=2, full line:T_r=3) and Of in PoF

Comparison with Related Methods. We compare our best results to several related methods. For HMDB51, we use HC/RC+F-OL+LPCA with the same setting before. For VOC2007, we use RC+F-OL+LPCA where the setting of SIFT is (N_r, T_r, S_r) = (20w×2×128) for RC; (D_{f1}, D_{f2}, O_f, S_f) = (80, 100, 80, 256) for F-OL. In Table 5, our approach outperforms the related methods on both datasets.

5. CONCLUSION

This paper proposes three novel methods to enhance codebook for VLAD representation. Compared to the traditional codebook generated by k-means, our methods are more robust to poor clustering of k-means, and effectively exploit the local structures of feature subspaces and data manifold. Our experiments show that all our methods achieve superior recognition accuracy than baselines.

6. ACKNOWLEDGEMENT

This work is partly supported by National Natural Science Foundation of China (91320101, 61472410, 61502470), Guangdong Innovative Research Program (2015B010129013, 201001D0104648280, 2014B050505017) and Shenzhen Research Program (KQCX2015033117354153, CYJ201304021-13127496).

7. REFERENCES

[1] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray, "Visual categorization with bags of keypoints," in *ECCVW*, 2004.

[2] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid, "Aggregating local image descriptors into compact codes," *PAMI*, 2012.

[3] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.

[4] Zhe Wang, Limin Wang, Wenbin Du, and Yu Qiao, "Exploring fisher vector and deep networks for action spotting," in *CVPRW*, 2015.

[5] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao, "Towards good practices for very deep two-stream convnets," *ArXiv:1507.02159*, 2015.

[6] Limin Wang, Zhe Wang, Yuanjun Xiong, and Yu Qiao, "CUHK&SIAT submission for thumos15 action recognition challenge," in *CVPR, THUMOS Challenge*, 2015.

[7] Limin Wang, Yu Qiao, and Xiaoou Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *CVPR*, 2015.

[8] Limin Wang, Zhe Wang, Wenbin Du, and Yu Qiao, "Object-scene convolutional neural networks for event recognition in images," in *CVPRW*, 2015.

[9] Limin Wang, Zhe Wang, Sheng Guo, and Yu Qiao, "Better exploiting os-cnns for better event recognition in images," in *ICCVW*, 2015.

[10] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010.

[11] Jonathan Delhumeau, Philippe Henri Gosselin, Hervé Jégou, and Patrick Pérez, "Revisiting the VLAD image representation," in *ACM MM*, 2013.

[12] Xiaojiang Peng, Limin Wang, Yu Qiao, and Qiang Peng, "Boosting VLAD with supervised dictionary learning and high-order statistics," in *ECCV*, 2014.

[13] Hervé Jégou, Matthijs Douze, and Cordelia Schmid, "Product quantization for nearest neighbor search," *PAMI*, 2011.

[14] Limin Wang, Yu Qiao, and Xiaoou Tang, "Motionlets: Mid-level 3d parts for human motion recognition," in *CVPR*, 2013.

[15] Andrea Vedaldi and Brian Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," in *ACM MM*, 2010.

[16] Mihir Jain, Herve Jegou, and Patrick Bouthemy, "Better exploiting motion for better action recognition," in *CVPR*, 2013.

[17] Olga Russakovsky, Yuanqing Lin, Kai Yu, and Fei-Fei Li, "Object-centric spatial pooling for image classification," in *ECCV*, 2012.

[18] Ken Chatfield, Victor S. Lempitsky, Andrea Vedaldi, and Andrew Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, 2011.

[19] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre, "HMDB: A large video database for human motion recognition," in *ICCV*, 2011.

[20] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman, "The pascal visual object classes challenge 2007 results," 2008.

[21] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *ArXiv:1405.4506*, 2014.