Boosting VLAD with Double Assignment using Deep Features for Action Recognition in Videos

Ionut C. Duta
University of Trento, Italy
ionutcosmin.duta@unitn.it

Tuan A. Nguyen
University of Tokyo, Japan
t_nguyen@hal.t.u-tokyo.ac.jp

Kiyoharu Aizawa
University of Tokyo, Japan
aizawa@hal.t.u-tokyo.ac.jp

Bogdan Ionescu
University Politehnica of Bucharest, Romania
bionescu@imag.pub.ro

Nicu Sebe
University of Trento, Italy
niculae.sebe@unitn.it

***Abstract*—The encoding method is an important factor for an action recognition pipeline. One of the key points for the encoding method is the assignment step. A very widely used super-vector encoding method is the vector of locally aggregated descriptors (VLAD), with very competitive results in many tasks. However, it considers only hard assignment and the criteria for the assignment is performed only from the features side, by looking for which visual word the features are voting. In this work we propose to encode deep features for videos using a double assignment VLAD (DA-VLAD). In addition to the traditional assignment for VLAD we perform a second assignment by taking into account the perspective from the codebook side: which are the nearest features to a visual word and not only which is the nearest centroid for the features as the standard assignment. Another important factor for the performance of an action recognition system is the feature extraction step. Recently, deep features obtained state-of-the-art results in many tasks, being also adopted for action recognition with competitive results over hand-crafted features. This work includes a pipeline to extract local deep features for videos using any available network as a black box and we show competitive results including the case when the network was trained for another task or another dataset. Our DA-VLAD encoding method outperforms the traditional VLAD and we obtain state-of-the-art results on UCF50 dataset and competitive results on UCF101 dataset.**

***Index Terms*—Video Classification, Action Recognition, Double Assignment VLAD (DA-VLAD), Deep Features.**

I. Introduction

Action recognition has attracted a considerable amount of attention in the computer vision community due to the ever increasing interest in the video processing and analysis applications, such as video indexing and retrieval, video surveillance, automatic recognition, etc. Even though in the past several years we have witnessed an important progress in action recognition research [22, 24, 16, 8], it is still an open issue mainly due to the difficulty of the data, namely: large intra-class variations, viewpoint changes, background clutter, high dimension of video data, low video resolution.

The Bag of Visual Words (BoVW) framework with its variations [11, 22, 24] has been widely used and showed its effectiveness in human action recognition challenges. The general BoVW pipeline to recognize actions contains several main steps: feature extraction and feature encoding followed by classification. Recently, the approaches based on convolutional neural networks (CNNs) [8, 17, 26, 25] have shown to obtain very competitive results related to traditional hand-crafted features, such as Histogram of Oriented Gradients (HOG) [3, 11], Histogram of Optical Flow (HOF) [11] and Motion Boundary Histograms (MBH) [4]. In general for action recognition in videos the CNNs based approaches use a two-stream approach where two networks are trained. The first network is trained on the raw frames of the video to capture the appearance information, and the second network is trained on the optical flow fields for capturing the motion information. In the end these two networks are combined to get the final score.

The feature encoding method is one of the key factors for visual recognition such as human action recognition. We can see that in most of the research works in computer vision [24, 14, 20, 21] the super-vector based encoding methods are shown to outperform the other encoding methods. Vector of locally aggregated descriptors (VLAD) is one of the most popular and efficient encoding methods which proved its performance in creating the final representation of a video for action recognition tasks. There are many precursors who focus on improving VLAD representation. The work in [12] proposes to use Random Forests in a pruned version for the trees to build the vocabulary and then they additionally concatenate second-order information, similar as in Fisher Vector. Another recent work which boosts the performance of VLAD is presented in [13]. They improve it by concatenating the second- and third-order statistics and using supervised dictionary learning. The work in [2] proposes to use intra-normalization to improve VLAD performance.

One of the key points for the encoding method is the assignment step. Besides the impressive performance, VLAD has several drawbacks, such as the assignment step. VLAD considers only hard assignment and the decision for the assignment is performed only from the features side, by looking for which visual word the features are voting.

In this paper we propose an enhanced approach to encode the deep features using double assignment for VLAD. Our second assignment brings complementary information to the first traditional assignment by considering also the perspective

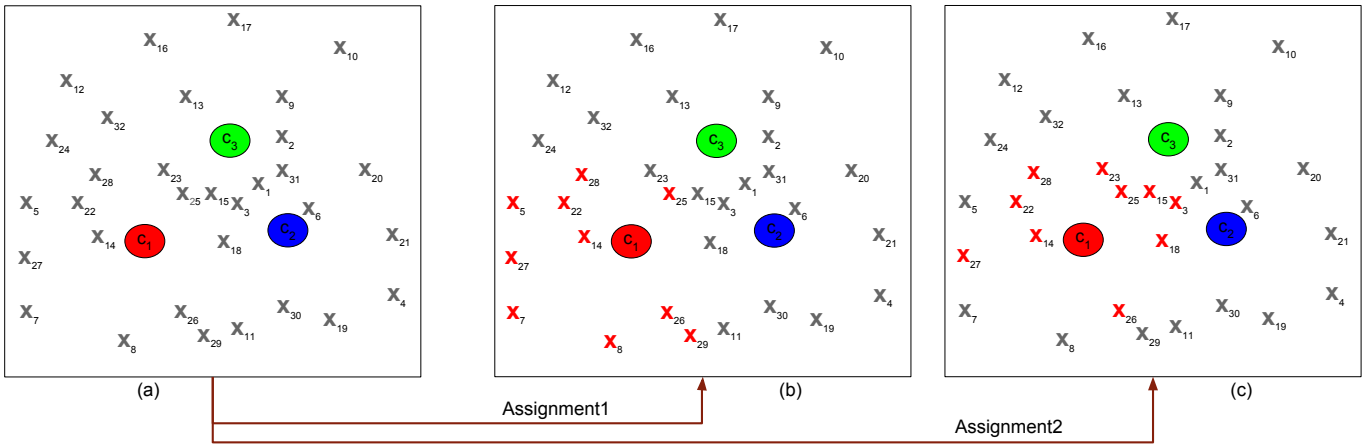

Fig. 1: Double assignment approach. (a) the features and centroids before assignment. (b) First assignment for the centroid C_1, as in standard VLAD, it receives N_1 features for which the closest visual word is C_1 (in this case N_1=10). (c) Second assignment for C_1, which assigns the closest N_i features to it.

from the centroids side: which are the nearest features for a visual word and not only which is the closest centroid as in standard assignment. The intuition behind this idea can be seen as principle that is checked in daily life: "not only how close you are to me is important but also how close I am to you". By looking only from one side to establish the closeness, a significant part of information is ignored. We denote our method the double assignment VLAD (DA-VLAD). Besides the way the assignment is carried out, another aspect of our approach is the proposed framework for action recognition, which can be used for dealing with deep features extracted from any already trained network. In our work we show that even if the network was not trained specifically for action recognition task, our approach is able to obtain very competitive results. Also, in this paper we present the cases when the network was trained and used for feature extraction on the same dataset, and when the network was trained on one dataset and used for feature extraction on a different dataset.

The contributions of this work can be summarized with the following:

- We propose a double assignment approach for VLAD which outperforms the baseline and obtain state-of-the-art result of action recognition.
- We propose a pipeline for feature extraction and feature encoding that can be easily adopted for any available network without the need to re-train the network.

The rest of the paper is organized as follows. The DA-VLAD encoding method is described in Section II. In Section III is illustrated the pipeline for deep feature extraction. In Section IV the experimental evaluation is presented. The conclusions are drawn in Section V.

II. Double Assignment VLAD

In this section we present our assignment approach for VLAD. The encoding method VLAD is one of the most used approachs and proved its efficiency in many tasks, including in action recognition. VLAD can be seen as a simplification version of Fisher Vectors (FV) and is initially proposed in [7]. The first step for VLAD is to learn a codebook of k visual words with k-means $C = \{c_1, c_2, ..., c_k\}$, which are the means for each cluster. Each local descriptor x_j from the set $X = \{x_1, x_2, ..., x_n\} \in R^{n \times d}$ is assigned to its nearest visual word. In the traditional VLAD the idea is to accumulate for each visual word the residuals (the differences between the assigned descriptors and the visual word). Instead of doing sum pooling of residuals as in the standard VLAD, we perform average pooling:

$$v_i = \frac{1}{N_i} \sum_{j=1}^{N_i} (x_j - c_i) \tag{1}$$

where N_i is the number of descriptors assigned to the cluster c_i. This division by the number of descriptors, that switches sum pooling to average pooling, is a very simple technique to deal with the problem of burstiness when some parts of VLAD can dominate the entire representation. In the rest of the paper we refer to VLAD as presented in this version.

Our approach, double assignment VLAD (DA-VLAD), is presented in an illustrative example in Fig. 1. After the vocabulary is created with k-means, in Fig. 1(a) are represented the resulted centroids c_1, c_2 and c_3. Suppose that for a given video the features are $\{x_1, x_2, ..., x_{32}\}$. We perform two approaches for assignment. First is the standard assignment procedure for VLAD, illustrated in Fig. 1(b), where, each local descriptor x_i is associated to its nearest visual word. Basically the features vote only for the nearest visual word, and for the traditional VLAD, the Equation 1 is computed only among these features for a specific cluster. In the case of the centroid c_1 the features assigned are $\{x_5, x_7, x_8, x_{14}, x_{22}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}\}$, thus, for c_1 the number of features assigned is $N_1 = 10$.

For the first assignment, by looking for which centroid the features are voting, we take into account only the perspective from the features side. By doing this some information is ignored. Complementary to the first assignment approach, the second assignment considers the other perspective, from the centroids side. After the first assignment, we have obtained the distances

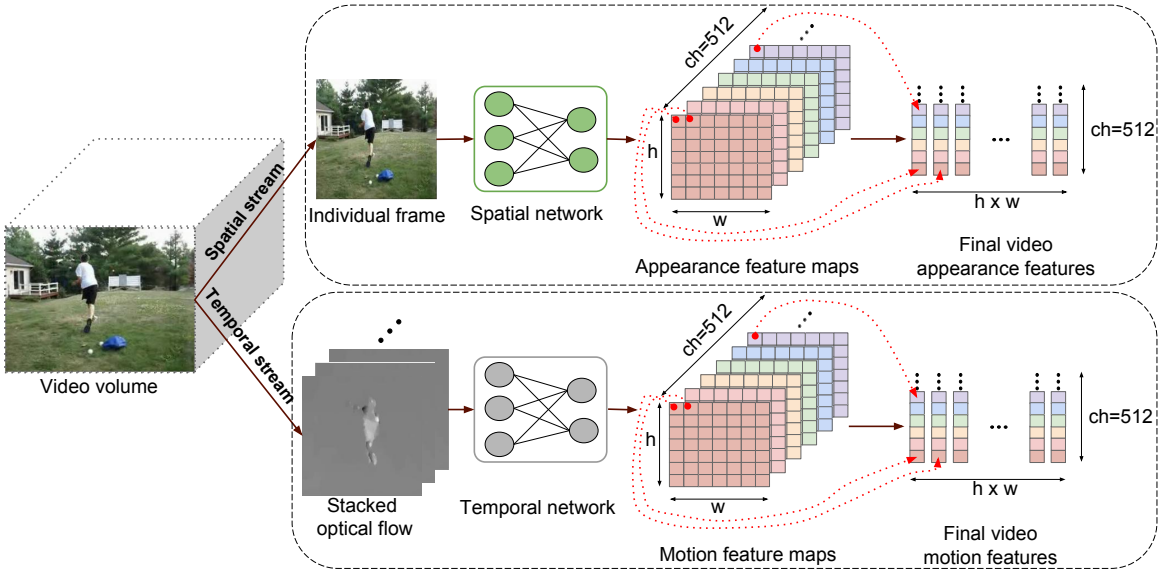

Fig. 2: Two-stream deep feature extraction pipeline for video classification.

between the features and the visual words and also the subset of features assigned to each centroid (and of course the size of each subset N_i).

For the second assignment, we sort the distances in the order of the nearest features to the centroids. After this step each visual word has the list of features ordered based on the nearest criterion. From these lists of ordered features we consider in our algorithm only the first N_i features for each visual word. We do not recommend to consider more than N_i features for the second assignment as over this number (which is specific for each centroid) the features are starting to be distant from the centroid c_i being not representative as information. In Fig. 1(c) is illustrated an outcome for the second assignment. In the case of the visual word c_1 the nearest 10 (as $N_1 = 10$) features assigned are $\{x_{14}, x_{22}, x_{25}, x_{18}, x_{23}, x_{15}, x_{28}, x_3, x_{26}, x_{27}\}$.

The final list of the features considered for double assignment is created by concatenating the list of assigned features from the first list with the assigned features from the second assignment. Finally, the VLAD formula (1) is applied on this concatenated lists of features to compute the final representation. The double assignment approach operates in two important directions. First, it considers twice the nearest features to the visual word. In the lists of the first and second assignment there are several common features, such as x_{14} or x_{22} in our graphic example. By combining the lists of both assignments the common features are considered twice. The effect of this approach is beneficial because the nearest features to the centroids receive a higher weight in the final resulted representation as they are very representative for the visual word.

The second important direction is that with this approach new features are considered, such as x_{18} or x_{23}. These kinds of features are also introduced due to the perspective from the centroid side, even if they did not initially vote for that centroid. These features are usually at the border between two centroids, and there are many cases when features that are farther are considered as belonging to the centroid and these features are not considered even if they are closer. For instance, the feature x_7 is farther to the visual word c_1 than the feature x_{18}, however, the first assignment considers only x_7 when computing the representation for the centroid c_1. For the second assignment we correct this aspect and consider x_{18} for computing the VLAD representation.

An important aspect for the second assignment is proportion of the features considered related to the number of the features considered at the first assignment. For instance, in the case of centroid c_1, for the first assignment we have 10 features considered and in our example for the second assignment we take also first 10 nearest features to the visual word c_1, therefore, in our example we considered 100% of the proportion for new features added by the second assignment. However, it is important to tune the proportion parameter Pr and to see the evolution for the cases between 0% and 100%. We present the tuning parameter Pr in the experimental section.

III. Deep Feature Extraction

For this work we consider deep features for action recognition in videos. Using deep features is a relatively new trend in action recognition with very promising results, and in several works it is presented with state-of-the-art performance over hand-crafted features. In this section we present the process of extraction deep features for videos. There are two important sources of information within the videos: appearance information contained in individual frames and motion information between each two consecutive frames. Therefore, the pipeline for action recognition follows this two directions.

Deep feature extraction step considers two-stream approach. Fig. 2 presents the framework to extract deep features for each stream. The spatial stream has the target to capture appearance information in the individual frames of the video. The first step of the spatial stream is to extract the individual frames from the video and the resize them to the requested input size for the

network of $224 \times 224 \times 3$. For spatial stream we use the VGG network with 19 layers of [17] trained on ImageNet dataset [5] for large-scale image recognition task. We choose this network due to its state-of-the-art performance, and also we show that our framework can be easily adapted to work with any network, and can obtain very competitive results using the network as a black box to extract features without the need for fine-tuning the network. For each individual frame that represents the input for the network we save the output of pool5, as is the last layer of the network which contains spatial information. The pool5 layer provides the feature maps for 512 channels (ch). The spatial size ($h \times w$) for each channel is 7×7. To obtain the local deep feature we concatenate the values for each spatial location along all the 512 channels. Therefore, the resulted local deep feature has the dimensionality of 512, which is equal to the number of the channels and the number of features for a frame is equal to the spatial size of the feature maps ($7 \times 7 = 49$ features per frame). After we obtain the local deep features for all the frames of a video, the pipeline for action recognition continues with the encoding steps of these features.

For the temporal stream to extract the deep features we use the re-trained network of [25], which re-trained the network VGG of [17] with 16 layers for a different task with different data. The authors present several good practices for the network re-training, such as pre-training to initialize the network, smaller learning rate, more date augmentation techniques and high dropout ratio. The input for the temporal network is 10-frame stacked of optical flow fields ($224 \times 224 \times 20$). To extract optical flow fields we use the OpenCV implementation of TVL1 algorithm [27]. After we obtain the optical flow fields images we resize them to 224×224 and input to the network 10 stacked optical flow fields (in total are 20 images for one input for the network). We save also the output of the last layer with spatial information (pool5), which contains also 512 channels with spatial size 7×7. For each spatial location we concatenate the output along the channels to create the local deep features from motion information as presented in Fig. 2. The total number of local deep features for a video in this case is: $7 \times 7 \times (\#frames - 9)$

IV. Experimental Evaluation

After we extract the deep features with the pipeline presented in the previous section we perform RootSIFT [1] normalization and then we apply PCA to reduce the dimensionality by a factor of two and decorrelate the features. This yields a final feature dimension of 256. The codebook necessary for feature encoding is built from randomly sampled 500K features. The number of visual words for the codebook is 256 as default setting. For the resulted vectors after feature encoding we apply power normalization followed by L2 normalization to have unit length of the video representation and then compute the distances between the video representations by inner product. The parameter α for power normalization is fixed to 0.5. We perform the classification using a linear one-vs-all SVM with $C = 100$.

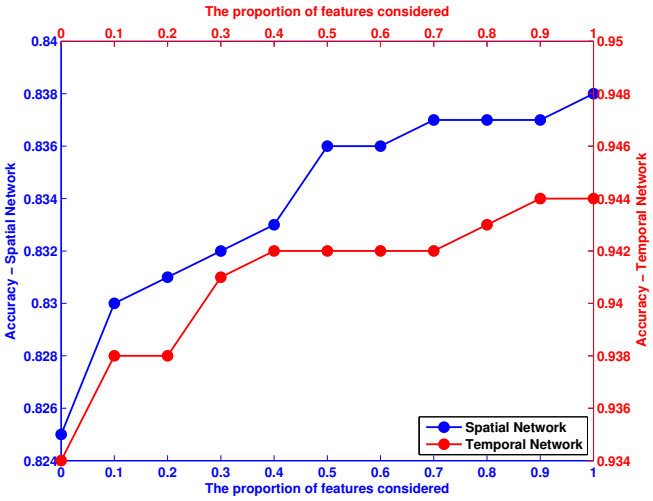

Fig. 3: Double assignment impact on the performance of the encoding method on UCF50 dataset.

A. Datasets

We evaluate our approach on UCF50 [15] and UCF101 [19] Human Action Recognition datasets. In total UCF50 dataset contains 6,618 realistic videos taken from YouTube with large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. There are 50 human action categories mutually exclusive, which range from general sports to daily life exercises. The videos are split into 25 predefined groups. We follow the recommended standard procedure and perform leave-one-group-out cross validation and report average classification accuracy over all 25 folds.

The UCF101 dataset is a widely adopted benchmark for action recognition, consisting in 13,320 realistic videos, which are divided in 25 groups for each action category. This dataset contains 101 action classes and there are at least 100 video clips for each class. We follow for evaluation the recommended default three training/testing splits. We report the average recognition accuracy over these three splits.

B. Parameter tuning and comparison to baseline

In this section we investigate the performance of our double assignment VLAD (DA-VLAD) and compare it with the VLAD representation. As pointed in the section above we use VGG network with 19 layers for the spatial network [17], for both UCF50 and UCF101. For the temporal network we use the re-trained networks of [25] corresponding to the three splits of UCF101, and for UCF50 we use the re-trained network for split1 of [25] . The first set of experiments that we present is the tuning parameter of the proportion (Pr) for the features considered on the second assignment. In the illustrative example of Fig. 1 we presented the case when we consider the first N_1 features (in that case N_1=10), therefore we consider an equal number of features as for the first assignment and the proportion in this 100%. The graph in Fig. 3 presents the cases when the proportion of the first nearest features to the centroid considered for the second

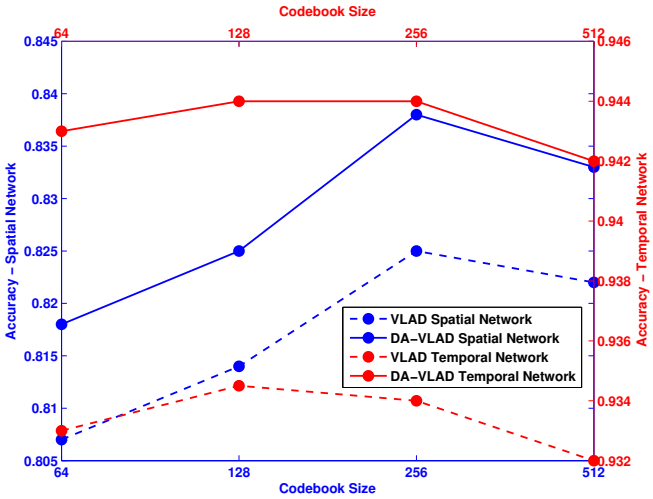

Fig. 4: Exploration of the number of visual words

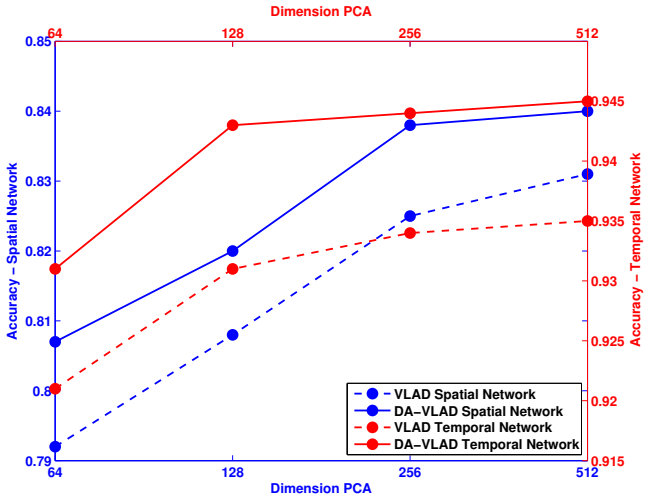

Fig. 5: Exploration of the dimension for PCA.

TABLE I: Final results for the Spatial Network

	UCF50	UCF101
VLAD Acc.	0.825	0.738
DA-VLAD Acc.	**0.838**	**0.754**

TABLE II: Final results for the Temporal Network

	UCF50	UCF101
VLAD Acc.	0.934	0.828
DA-VLAD Acc.	**0.944**	**0.839**

assignment is less than the number of features considered for the first assignment. A 0 value for the proportion of features considered in the second assignment leads to the classic VLAD representation.

By considering for the second assignment only the first 10% of the nearest features related to feature number of the first assignment, the accuracy is boosted from 0.825 to 0.830 for spatial network and from 0.934 to 0.938 when using temporal network. From the graph we can notice that by considering more features for the second assignment the accuracy is improved in the case of both networks. The drawback of our method is that for the best results the proportion parameter of the features considered for the second assignment should be tuned for each system used. For the remaining experiments we set as default the proportion to 1 between first and second assignment, therefore for the second assignment we will consider an equal number of features with the first assignment.

Fig. 4 presents the evolution graph when the number of visual words are changed and the other parameters are fixed to the default values pointed above. We can see from the graph that the size of the codebook of 256 gives the best trade-off for both temporal and spatial networks. For the rest of the paper we keep the codebook size to 256 visual words as default value.

The graph in Fig. 5 illustrates the evolution of the accuracy when the dimension of PCA is changed. The best results for accuracy are obtained when we keep all 512 dimension of the features. However, the increasement from 256 to 512 is not significant considering that the dimensionality is two times higher. Considering this, we set the dimension of PCA to 256 as this value is a good trade-off between accuracy and feature dimensionality.

The final results for UCF50 and UCF101 of our DA-VLAD encoding method compared with the baseline VLAD are presented in Table I for the spatial network. The encoding method DA-VLAD boosts the performance from 0.825 to 0.838 for UCF50 and from 0.738 to 0.754 for UCF101. The results for temporal network are presented in Table II where we can see for both datasets our DA-VLAD approach outperforms the baseline.

C. *DA-VLAD two-stream*

For combining the spatial network with the temporal network we use early fusion of the resulted video representation after encoding. Before fusion of the both representations of capturing appearance information respectively motion information, each of them follow independently the pipeline for feature extraction and feature encoding with DA-VALD, using the default settings: RootSIFT normalization before PCA, 256 PCA dimension, 256 visual words for codebook and the proportion of the features considered for the second assignment equal to 1.

Before the concatenation of the both video representations resulted after DA-VLAD we perform separately power normalization ($\alpha = 0.5$) followed by L2 normalization. After the concatenation we make unit length of the final representation by applying L2 normalization and then we apply linear SVM with $C = 100$. The fusion of spatial stream with temporal stream boosts the performance to 0.954 for UCF50 and to 0.886 UCF101. The comparison of our final results with other state-of-the-art approaches is presented in the next subsection.

D. *Comparison to state-of-the-art*

Table III presents the comparison with state-of-the-art approaches for UCF50 dataset. The proposed approach DA-VLAD with two-stream obtains state-of-the-art results outperforming including the recent work of [23] which considers as encoding method the spatial FV [10] together with spatio-temporal pyramid [11]. Our results are better than [14] which considers

TABLE III: Comparison to the state-of-the-art for UCF50 dataset.

Method	Accuracy
Kliper-Gross et al. [9] (2012)	0.727
Solmaz et al. [18] (2012)	0.737
Reddy et al. [15] (2012)	0.769
Uijlings et al. [21] (2014)	0.818
Wang et al. [22] (2013)	0.856
Wang et al. [24] (2013)	0.912
Wang et al. [23] (2015)	0.917
Peng et al. [14] (2014)	0.923
Duta et al. [6] (2016)	0.930
DA-VLAD two-stream	**0.954**

TABLE IV: Comparison to the state-of-the-art for UCF101 dataset.

Method	Accuracy
Karpathy et al. [8] (2014)	0.654
Wang et al. [24] (2013)	0.859
Wang et al. [23] (2015)	0.860
Peng et al. [14] (2014)	0.879
Simonyan et al. [17] (2014)	0.880
Ng et al. [26] (2015)	0.886
Wang at al. [25] (2015)	**0.914**
DA-VLAD two-stream	0.886

a hybrid representation by combining two different representations. Our approach outperforms also the very recent work of [6], which proposes a new spatio-temporal descriptor for capturing the motion information without the need of computing the expensive optical flow fields, in the end the authors combine the proposed descriptor with Improved Trajectories of [24] obtaining competitive results.

The comparison with state-of-the-art for UCF101 dataset is presented in Table IV. Our framework outperforms all the hand-crafted features based approaches, including hybrid representation of [14] and the recent work of [23]. Our approach obtains better performance than many popular approaches based on convolutional neural networks, such as [8, 17, 26]. The approach based on convolutional neural networks of [25] obtains state-of-the-art results on UCF101, however, the advantage of our framework is that it can be adapted to any convolutional neural network without the need to re-train the network.

V. Conclusion

In this work we introduce the double assignment VLAD (DA-VLAD), which boosts the performance of VLAD by considering a complementary second assignment that takes into account the assignment from the perspective of the visual words. The experimental results show that our proposed approach outperforms the baseline VLAD. Furthermore, we present a pipeline to extract local deep features that can be used for any network. We show the case in this paper when the network is trained for another task with different data, and with our pipeline very competitive results for action recognition can be obtained. The proposed framework obtains state-of-the-art result on UCF50 dataset and competitive results on UCF101 dataset. For future extension of this work we will focus on combining the deep features with hand-crafted features under this framework for boosting further the performance of action recognition.

VI. Acknowledgments

This work has been supported by the EC FP7 project xLiMe.

References

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[2] R. Arandjelovic and A. Zisserman. All about VLAD. In *CVPR*, 2013.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*. 2006.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[6] I. C. Duta, J. R. R. Uijlings, T. A. Nguyen, K. Aizawa, A. G. Hauptmann, B. Ionescu, and N. Sebe. Histograms of motion gradients for real-time video classification. In *CBMI*, 2016.

[7] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716, 2012.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[9] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[10] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *ICCV*, 2011.

[11] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[12] I. Mironică, I. C. Duţă, B. Ionescu, and N. Sebe. A modified vector of locally aggregated descriptors approach for fast video classification. *Multimedia Tools and Applications*, in press, 2016.

[13] X. Peng, L. Wang, Y. Qiao, and Q. Peng. Boosting vlad with supervised dictionary learning and high-order statistics. In *ECCV*. 2014.

[14] X. Peng, L. Wang, X. Wang, and Y. Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *arXiv:1405.4506*, 2014.

[15] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[16] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] B. Solmaz, S. M. Assari, and M. Shah. Classifying web videos using a global video descriptor. *Machine vision and applications*, 2013.

[19] 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.

[20] J. R. R. Uijlings, I. C. Duta, N. Rostamzadeh, and N. Sebe. Realtime video classification using dense hof/hog. In *ICMR*, 2014.

[21] J. R. R. Uijlings, I. C. Duta, E. Sangineto, and N. Sebe. Video classification with densely extracted hog/hof/mbh features: an evaluation of the accuracy/computational efficiency trade-off. *International Journal of Multimedia Information Retrieval*, 4(1):33–44, 2014.

[22] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.

[23] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *IJCV*, 2015.

[24] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[25] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015.

[26] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[27] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*. 2007.