REJECTION-BASED CLASSIFICATION FOR ACTION RECOGNITION USING A SPATIO-TEMPORAL DICTIONARY

Stefen Chan Wai Tim, Michele Rombaut, Denis Pellerin

Univ. Grenoble Alpes, GIPSA-Lab, F-38000 Grenoble, France

ABSTRACT

This paper presents a method for human action recognition in videos which learns a dictionary whose atoms are spatio-temporal patches. We use these gray-level spatio-temporal patches to learn motion patterns inside the videos. This method also relies on a part-based human detector in order to segment and narrow down several interesting regions inside the videos without a need for bounding boxes annotations. We show that the utilization of these parts improves the classification performance. We introduce a rejection-based classification method which is based on a Support Vector Machine. This method has been tested on UCF sports action dataset with good results.

Index Terms— Dictionary Learning, Action Recognition, Classification, Videos, Spatio-temporal patches

1. INTRODUCTION

The research on action recognition has developed a lot in the last years along with the rise of video contents, especially because its applications are numerous in surveillance, automatic video annotations or entertainment. Generally, it consists in extracting features either globally (or in successive frames) [1, 2] or locally [3]. The goal is to classify some human activities using the data extracted from videos.

Techniques based on dictionaries and sparse representations [4–6] have emerged in action recognition because of the possibility to represent complex patterns using a few dictionary elements. These methods rely on the creation of a dictionary which can encode effectively the information contained in an image.

In this paper, we propose a method for human action recognition in the context of gray-level video sequences. This method learns a dictionary of spatio-temporal atoms coupled with a part-based human detector to select interesting spatio-temporal regions inside a video. We also introduce an original three-stepped rejection-based classification method based on a SVM in order to improve the results. The paper is organized as follows: Section 2 presents the proposed action recognition method, Section 3 describes the results obtained and Section 4 presents the conclusions.

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025).

2. METHOD FRAMEWORK

The proposed method consists in classifying temporal blocks composed of several frames and containing a human who perfoms one particular action which corresponds to a class.

We call *temporal block* t successive frames of a video. In these blocks, we define *spatio-temporal volumes* which correspond to the same spatial regions taken in successive frames inside the videos. These spatio-temporal volumes are localized around areas of interest. Spatio-temporal volumes themselves are composed of small *spatio-temporal patches* which are defined as a succession of spatial image patches (see Figure 1). A dictionary is a learned collection of such patches called *atoms*.

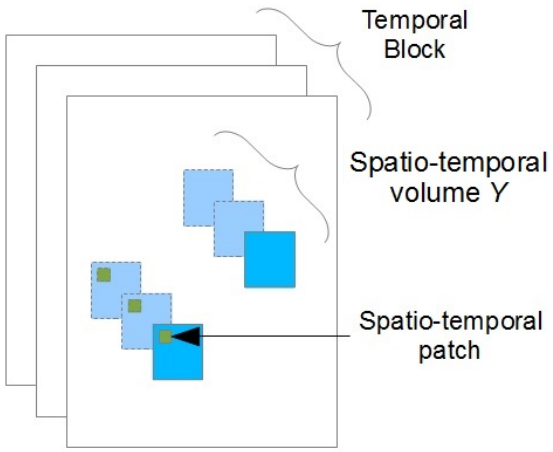


Fig. 1. Example of a temporal block of $t = 3$ frames described with 2 spatio-temporal volumes. Each volume is composed of a collection of small spatio-temporal patches.

The method described below is based on a previous work [7] and relies on a spatio-temporal patch-based dictionary to represent spatio-temporal volumes. In this paper, we also use a part-based human representation to narrow down the image

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

to interesting regions. This method is described in 3 parts: (i) Dictionary learning, (ii) Spatio-temporal objects and (iii) Classification with rejection.

2.1. Dictionary Learning

At first, dictionary learning methods were used for denoising and inpainting [8, 9] and now, different methods exist in the literature to learn dictionaries for sparse representations. Some of them are specifically designed for classification, for example DKSVD [10] or GDDL [11]. Dictionary-based methods got great success, in particular, in face recognition applications. However, these methods are often application specific and cannot be straightforwardly used in action recognition because they rely on the alignment of the images.

We want to learn a dictionary on spatio-temporal patches in order to describe movement patterns. We cannot utilize large patches and learn a dictionary that can efficiently discriminate between the different classes because the actions are complex and we cannot align these images. As a consequence, we are using small patches compared to the spatio-temporal volumes. Since the patches are small compared to the complexity of the data, there may be no specific or discriminative atoms for each class (different classes may utilize similar atoms) and the recognition is done solely on the proportion of used atoms. That is why we chose to implement K-SVD [9] for learning the dictionary.

Let $\mathbf{p}$ be a spatio-temporal gray-level patch of size $(s \times s \times t)$, with s being the size of a square patch in the spatial dimensions and t being the number of frames considered in the temporal dimension. The patch $\mathbf{p}_{norm}$ is the patch $\mathbf{p}$ whose set of pixels are normalized so that it has zero mean and unit norm.

Let $Y = [\mathbf{p}_{norm,1}, \mathbf{p}_{norm,2}, ..., \mathbf{p}_{norm,m}] \in \mathbf{R}^{n \times m}$ be a a matrix composed of spatio-temporal patches, with m being the size of the dataset, n the dimension of the patches and $D = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{N_{atoms}}] \in \mathbf{R}^{n \times N_{atoms}}$ be the dictionary of N_{atoms} atoms $\mathbf{d}_k$, where the number N_{atoms} is chosen empirically.

The dictionary learning algorithm K-SVD [9] is an iterative algorithm which can solve the optimization problem. The formulation of the dictionary learning algorithm is:

$$\min_{D,X} \{\|Y - DX\|_F^2\} \text{ such that } \forall i \in [1, m], \|\mathbf{x}_i\|_0 \leq T_0 \quad (1)$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m] \in \mathbf{R}^{N_{atoms} \times m}$ contains the coefficients of the decomposition of Y using the dictionary D. $\mathbf{x}_i = (\alpha_j)_{j \in [1, N_{atoms}]}$ is a column vector from X describing $p_{norm,i}$ and $\|\mathbf{x}_i\|_0$ is the norm that counts the number of non-zero entry of the vector. $\|\cdot\|_F$ is the Frobenius norm: $A \in \mathbf{R}^{n \times m}$, $\|A\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}|^2}$. T_0 is the maximum of non-zero entries.

This algorithm is performed in two steps: first by optimizing the codes X with a fixed dictionary and secondly by optimizing D with respect to X.

The output of this learning step is a dictionary D composed of N_{atoms} (Figure 2). Then, we can compute a sparse representation of a spatio-temporal volume Y based on the dictionary.

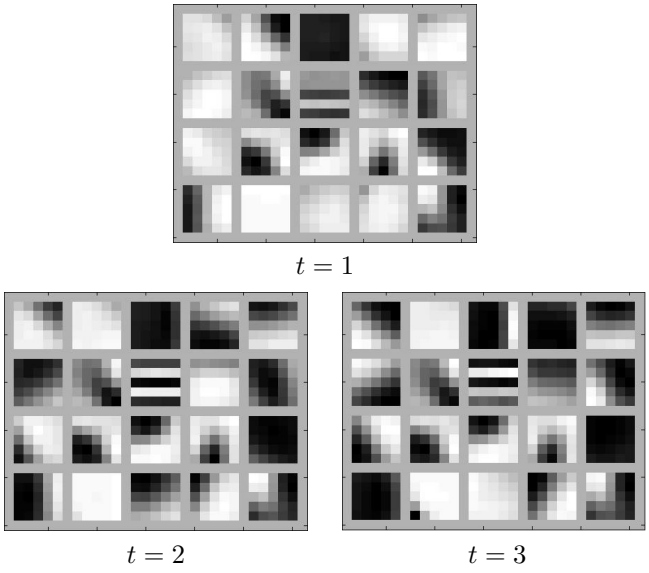


Fig. 2. Example of 20 atoms of size $(5 \times 5 \times 3)$ pixels from a learned dictionary with K-SVD.

2.2. Spatio-temporal objects

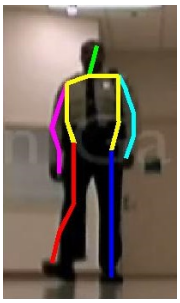
Our objective is to classify successive frames (temporal block) of the video. To do that, we propose to select interesting regions within the images (see Figure 1) based on the results of a part-based human detection algorithm.

2.2.1. Part-based human detection

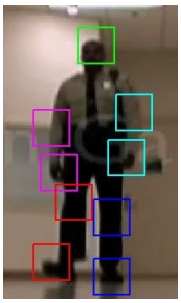
We perform the selection of interesting regions within the images thanks to an existing part-based human detection algorithm [12]. The algorithm is based on a deformable parts model capable of handling large variations in the human poses. The output of the algorithm is a list of square bounding boxes of fixed size corresponding to the localizations of the different parts of the model. In the model chosen, the parts are positioned on specific regions of the human body (head, right/left arm, right/left leg ...). The original part-based model contains 26 different parts and we selected only 9 because they approximatively cover the body (Figure 3) and in order to limit the dimension of the block representation (see Section 2.2.2).

From each of the parts, we construct a spatio-temporal volume Y (see Figure 1) by extending the localization of the

parts in the temporal dimension.



(a) Articulated human



(b) Selected boxes

Fig. 3. Example of bounding boxes extracted by [12]. (a) is an example of a full part-based human detection. (b) is an example of the 9 parts selected. The images are obtained from a video of the UCF dataset from the class "Walking".

2.2.2. *Signature computation*

We want to classify the human actions represented in videos using the spatio-temporal volumes described in the previous section.

Each volume Y can be described with a histogram:

$$h = \sum_{i=1}^{m} \mathbf{x}_i$$

with the coefficient vectors $\mathbf{x}_i$ of all the patches within Y. Since we are using 9 parts, we can obtain 9 different histograms $h^{(j)}$, $j = \{1, ..9\}$. The final signature is the concatenation of the histograms for all of the individual parts: $h_{block} = [h^{(1)}, ..., h^{(j)}]$. The final dimension of the signature is $N_{parts} \times N_{atoms}$.

2.3. Classification with rejection

Once we obtain the signatures for each temporal block, we want to retrieve the action classes for each video. The extracted features consist in signatures obtained for each block of t frames, meaning that, for a video of L frames containing a single action, we can get up to $L-t+1$ signatures if we use overlapping blocks of frames. We have developed a method that gives a label to each block. The label of the video is obtained by a voting procedure on all the blocks of the video, but since the dimension of the patches and the number of frames t taken into account in each block are small, some signatures may be ambiguous leading to classification errors. That is why we decided to add an extra "rejection" class which regroups these ambiguous signatures when training the SVM. Normally, all the signatures extracted from the blocks for the same video share the same label: as a consequence, the objective of this new class is to prevent some signatures that could be ambiguous from voting for a label.

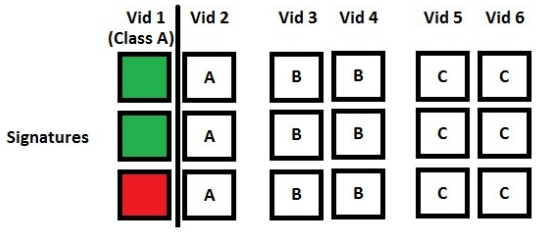
The SVM is trained in three steps.

In the first step, we divide the training set into groups where each *group* is the set of signatures belonging to a given video of L frames. Indeed, the signatures within the same video normally share a lot of similarities and thus cannot be processed separately. Then we do a leave-one-group-out setup on the training set to optimize the parameters of the SVM.

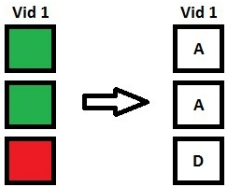
In the second step, we use the previous leave-one-group-out setup with the optimized parameters and we look for each signature misclassified in each group. The misclassified signatures are moved into the "rejection" class (see Figure 4).

In the third step, a final classifier is learned using all the signatures and all the classes (including the "rejection" class) as input. The final label for a given video in the test set is obtained by voting using the set of signatures of the videos. Each signature classified in the rejection class is removed from the vote.

We have found that this method can significantly improve the classification results. However, we have to carefully balance the number of elements in the "rejection" class after the second step. Otherwise, it happens that, during the testing phase, all the signatures of a video can end up classified in the "rejection" class. We tried different rules for moving an element into the "rejection" class. The results are presented in Section 3.



(a) One iteration of the leave-one-group-out in the first step. (green = well classified, red = misclassified)



(b) One signature of Video 1 is moved to "rejection" class D in the second step.

Fig. 4. Example of the second step of the proposed classification method with 6 videos and 3 classes of actions A, B and C. We use a Leave-one-group-out setup with the signatures of each video as a group. Each signature misclassified during this step goes into the "rejection" class. At the end of this step, the whole training set (signatures) with the extra class serves to learn the final classifier.

3. EXPERIMENTATIONS

We tested the proposed algorithm of the UCF sports action dataset [13]. This dataset is normally composed of 10 classes: "Walking", "Swinging", "Skateboarding", "Running", "Riding horse", "Lifting", "Kicking", "Playing golf" and "Diving". The resolution of the videos is (720×480) pixels. We decided to remove the class "Lifting" because of the lack of annotations. In total, we used 9 classes and 140 videos.

For the classification of the videos, we used libLinear [14]. We did a leave-one-group-out using all the signatures computed for a single video as a group. The size of the patches considered is $(5 \times 5 \times 3)$ pixels. With the selected temporal size $t = 3$, we obtained between 20 and 90 signatures for each video depending on its length. Each human was described with 9 volumes of $(24 \times 24 \times 3)$ pixels computed with the part-based detector. The classification label was determined by votes using all the signatures (or the signatures not classified in the "rejection" class). Table 1 shows the different performances with and without the use of the part-based detector and the "rejection" class. We reach 86.4% accuracy with 9 parts and a dictionary D of $N_{atoms} = 150$ atoms against 72.1% when considering only one full bounding box including all the body. We chose to take the sparsity $T_0 = 1$ for our experiments. In our previous work, we showed that taking any low value for T_0 did not change the results much but $T_0 = 1$ gave us the best performance. The choice of patch size is tied with the dictionary size for overcompleteness. A larger dictionary also leads to larger signature size. Empirically, we found that the choice of $(5 \times 5 \times 3)$ for the patch size was a good compromise.

We also tried learning a specific dictionary for each individual part used in our method but the results obtained were the same. Our hypothesis is that the global motion information of the parts is more important than the local precision in the shape. However, we can also add the precision of the part detection is not perfect for some classes (for example, because of the blur). Moreover, we note that even if many of the boxes are not well-localized, it still improves the classification performance by a good margin (see Table 1). We believe that, when using a single full bounding box, too many spatial information is lost during the pooling. The fact of using the part detector is a way to reintroduce some spatial information and narrowing down some interesting image regions.

We tried different rules for the "rejection" class. The first rule tested is to move the misclassified signatures during the first learning step into the "rejection" class. However, it happened that in some cases, for the test set, all the signatures for a given video could end up rejected meaning that we had to use the classifier learned without our extra class. To prevent that, we tried softer rules: instead of moving all misclassified signatures into the extra class, we only moved signatures with a probability to belong to its true class below a chosen value (see Table 2). These probabilies can be obtained by adding an option in libLinear and are computed using logistic regression for each class. We can see that our extra learning step serves its purpose since the accuracy at signature level jumps from 53.6% to about 80%. Moreover, we can also observe that balancing the "rejection" class can lead to improved results: going down from 50% rejected signatures to 35% leads to a gain of about 2% in accuracy because of the effects described above.

Methods	Accuracy
Gray-level + Full bounding boxes	72.1%
Gray-level + Part-based detector	80.0%
Gray-level + Part-based detector (SVM with basic "rejection" class)	86.4%

Table 1. Performance comparison for our method in different conditions. Dictionary size = 150, Sparsity $T_0 = 1$, video labels obtained by voting using the signatures of the video.

Rejection rule	Video Label Accuracy	Signature Accuracy	% Signatures rejected
Reference	80.0%	53.6%	
No Probability Estimates	86.4%	87.8%	50.4%
Probability Estimates (Thr: 0.25)	86.4%	87.5%	38.5%
Probability Estimates (Thr: 0.15)	88.6%	85.9%	35.5%
Probability Estimates (Thr: 0.05)	87.9%	83.9%	31.2%

Table 2. Performance comparison for different rejection rules. Dictionary size = 150, Sparsity $T_0 = 1$, video labels obtained by voting using the signatures of the video. The table gives the video label classification accuracy (obtained after voting), the signature classification accuracy and the proportion of signatures rejectedby the classification.

We compare the proposed method with different algorithms of the literature in the Table 3. We can see that our method achieves good results. Even if the descriptors obtained with the part-based detector alone only reach 80%, the combination of the proposed descriptors and classification method perform really well. The size of the codebook used in our method is 150 compared to 4000 for [15] and 40 for [16] even if the final dimension of the signatures is 1350. Speed wise, the experiments were done on MATLAB so a significant gain in speed is possible: the part-based human detector was the limiting factor with about 0.5 frame per second on a

modern computer, once the parts are extracted we can compute about 7 signatures per second. Results from [17, 18] are given as a comparison: they use techniques different from dictionary-based approach and show that the proposed method is competitive.

Methods	Accuracy
Dictionary based methods	
H. Wang et al. [15]	85.6%
Q. Qiu et al. [16]	83.6%
T. Guha et al. [5]	83.8%
Proposed method	88.6%
Other methods	
X. Wu et al. [17]	91.3%
H. Wang et al. [18]	88.0%

Table 3. Performance comparison for different features and methods for the classification on UCF Sports action dataset. Both dictionary-based and non dictionary-based methods are presented.

4. CONCLUSION

Given a raw video as input, the proposed dictionary-based action recognition method performs efficiently. Despite the small dimension of the patches, we show that using localized spatio-temporal volumes improves the results. We also introduced a rejection-based classification method to select the most descriptive signatures to select the labels. The method has been tested on UCF sports action dataset with good results.

For future work, we are looking for a way to take into account the evolution of the signatures extracted from successive temporal blocks instead of treating them separately. We are also working on more recent databases like UCF50.

REFERENCES

[1] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," *ICCV*, 2003.

[2] K. Schindler and L. V. Gool, "Action snippets: How many frames does human action recognition require?," *CVPR*, 2008.

[3] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," *Proceedings of the 15th international conference on Multimedia*, 2007.

[4] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," *CVPR*, 2009.

[5] T. Guha and R. K. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, 2012.

[6] G. Somasundaram, A. Cherian, V. Morellas, and N. Papanikolopoulos, "Action recognition using global spatio-temporal features derived from sparse representations," *CVIU*, 2013.

[7] S. Chan Wai Tim, M. Rombaut, and D. Pellerin, "Dictionary of gray-level 3d patches for action recognition," in *MLSP*, 2014.

[8] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. volume 11, pp. pages 19–60, 2010.

[9] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, 2006.

[10] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *CVPR*, 2010.

[11] Y. Suo, M. Dao, U. Srinivas, Vishal Monga, and T. D. Tran, "Structured dictionary learning for classification," *IEEE Transactions on Signal Processing*, 2014.

[12] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, 2013.

[13] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach: A spatio-temporal maximum average correlation height filter for action recognition," in *CVPR*, 2008.

[14] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, 2008.

[15] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *British Machine Vision Conference*, 2009.

[16] Q. Qiu, Z. Jiang, and R. Chellappa, "Sparse dictionary-based representation and recognition of action attributes," *ICCV*, 2011.

[17] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *CVPR*, 2010.

[18] H. Wang, A. Klaser, C. Schmid, and C-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," in *IJCV*, 2013.