Featureless

Bypassing feature extraction in action categorization

Pintea, Silvia; Mettes, Pascal; van Gemert, Jan; Smeulders, AWM

DOI
10.1109/ICIP.2016.7532346

Publication date
2016

Document Version
Accepted author manuscript

Published in
2016 IEEE International Conference on Image Processing (ICIP)

FEATURELESS: BYPASSING FEATURE EXTRACTION IN ACTION CATEGORIZATION

S. L. Pintea[a], *P. S. Mettes*[a]

[a]Intelligent Sensory Information Systems,
University of Amsterdam,
Amsterdam, Netherlands

J. C. van Gemert[a,b], *A. W. M. Smeulders*[a]

[b]Computer Vision Lab,
Delft University of Technology,
Delft, Netherlands

ABSTRACT

This method introduces an efficient manner of learning action categories without the need of feature estimation. The approach starts from low-level values, in a similar style to the successful CNN methods. However, rather than extracting general image features, we learn to predict specific video representations from raw video data. The benefit of such an approach is that at the same computational expense it can predict $2D$ video representations as well as $3D$ ones, based on motion. The proposed model relies on discriminative Waldboost, which we enhance to a multiclass formulation for the purpose of learning video representations. The suitability of the proposed approach as well as its time efficiency are tested on the UCF11 action recognition dataset.

Index Terms— Multiclass Waldboost, video representations, action recognition, feature learning.

1. INTRODUCTION

Ever since the bag-of-words representation of visual information was proposed [1], the focus has been on feature robustness [2, 3, 4]. The popular deep CNN (Convolutional Neural Networks) [5, 6, 7] have effectively replaced the handcrafted descriptors with network features. Such networks have been successfully applied in the domain of action recognition [8, 9, 10, 11]. More recently, CNN features are used together with Fisher Vectors to build stronger video representation [12, 13]. However, competitive performance for action recognition is still achieved by video representations relying on appearance and motion descriptors [14, 15, 16]. This work proposes a manner of learning a given video representation, rather than learning better features to be subsequently used in the video representation, as in the case of CNN features. Illustrated in figure 1 is the proposed method that bypasses feature computation and, instead, learns the final video representation.

This work presents a proof of concept which challenges the idea of discarding the feature estimation and learning in one step the transition from low-level data to the final video representation. The premise of this paper is to keep the advanced analysis as simple as possible, therefore, here we focus on the more simplistic bag-of-words model. We research how much of this classic pipeline can be discarded while still achieving comparable performance. In the proposed method we learn from low-level values to predict a known codebook assignment — thus, at test time neither the image descriptors, nor the codebook need to be defined. Motion features as well as appearance features together with their codebook assignments are bypassed in the featureless method.

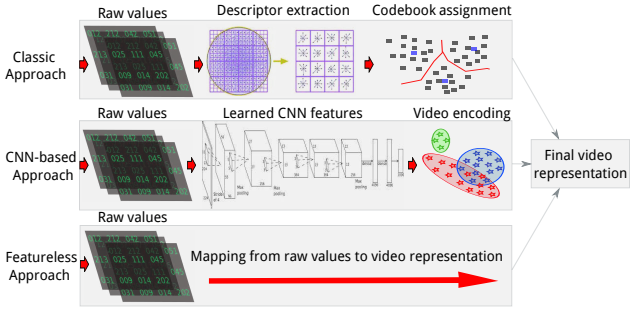

Fig. 1. Going from raw video values to higher level features. We propose bypassing feature computation by learning the mapping from raw input data to any type of higher order video representation — i.e. based on appearance, motion descriptors.

We rely on boosting to solve the learning problem. The gain of starting from boosting techniques is their well known quality of being specifically appropriate for real time applications [17, 18]. This is a desirable property in the context of videos. We start from individual values and build a strong classifier by incorporating multiple local cues in a boosting framework. In this work, we propose a straightforward discriminative multiclass extension of Waldboost [19]. This is employed to learn the mapping from the low-level input visual information to the desired representation.

Tasks such as large scale video categorization and event recognition still rely on the use of large codebooks and descriptor extraction [20, 21]. Moreover, the action recognition field deals with large amounts of data whose processing comes at considerable computational costs. This work can prove useful for such approaches, as it discards descriptor extraction and the need of codebooks or other video encodings such as Fisher Vectors at test time. Figure 2 displays the proposed framework in the context of action recognition.

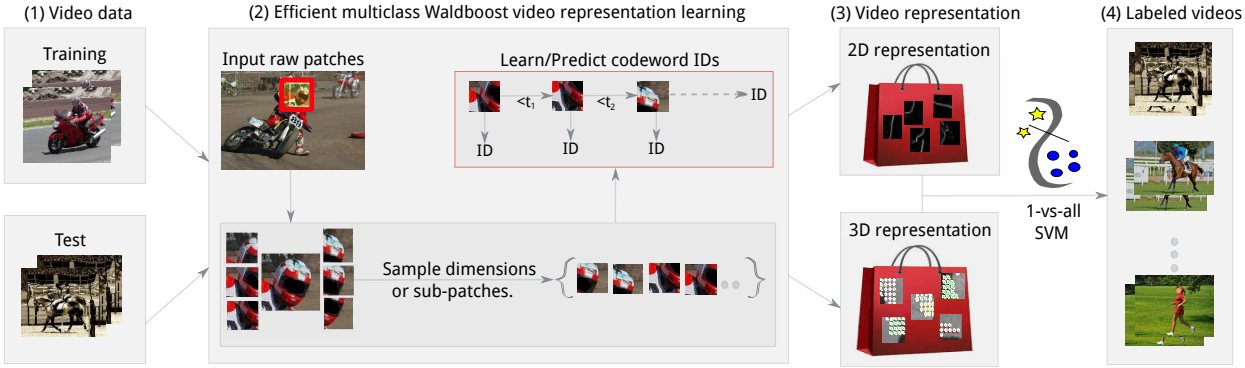

Fig. 2. We start from input videos, we subsequently extract frame patches to be used as training samples for the multiclass Waldboost. During training we estimate appearance and motion descriptors — HOG and HOF and $3D$ HOF — and we build a codebook which defines the training labels. For each low-level patch extracted from an input video frame, we learn its final codebook assignment in the proposed multiclass Waldboost extension. At test time we discard the codebook and predict in one step, from low-level patches, the final codebook assignment which is subsequently input to an SVM for the final video categorization.

2. RELATED WORK

In the literature the focus has been on either more compact and robust descriptors [2, 3, 4], or on stronger video encodings [22, 23, 24]. The effective CNN methods are able to produce strong image features [5, 6, 7]. Followed by a subsequent video encoding step, these methods represent the state-of-the-art in action recognition [12, 13]. In this work we do not focus on learning either the image/frame features or the video encoding, but instead, we focus on learning a time-efficient classifier that bypasses these steps and retrieves the final video representation.

We use as starting point the real-valued multiclass Adaboost [25]. This is extended in a discriminative manner to determine at each iteration whether to continue with the evaluation of the next weak classifier or stop. Thus, it allows for fast decision making as not all weak classifiers need to be employed. As described in section 3.1, this corresponds to a discriminative multiclass extension of Waldboost [19].

In the action recognition literature, efficiency has been a main focus [26, 27]. Proposed methods are ranging from faster features, based on faster flow computation, to faster video encodings, based on additive approximations. Methods such as [9, 11, 28, 29] have focused on improving the CNN architecture to achieve better performance in the context of action recognition. Yet, with the gain in performance comes also a gain in speed as deep neural networks are known to be fairly efficient at test-time. However, when focusing on the performance gain, methods [14, 15, 16] successfully rely on handcrafted visual descriptors such as HOG, HOF and MBH, and video encodings such as Fisher Vectors and VLAD. In this work we propose to combine the best of both worlds — allow to obtain video representations similar to the ones based on visual descriptors such as HOF, HOG and MBH while not having to estimate these descriptors or define the video encoding. Therefore, we bypass feature computation and video encoding and learn a direct mapping from low-level data.

3. LEARNING THE MAPPING

3.1. Multiclass Extension of Walboost

Data and Labels. We use grayscale pixel values as input to the boosting pipeline. Given and input image patch, we perform L_1 normalization over the patch. During the boosting training we sample randomly $\sqrt{D}$ dimensions, where D is the total number of data dimensions. For stability, we repeat the selection step $\sqrt{D}$ number of times and keep the feature dimensions that provide the best performance on the training data. We use these dimensions to train a weak classifier. During training, each patch of gray values is associated with a descriptor which is subsequently projected on a codebook in order to retrieve the codebook assignment. The codebook assignment defines the labeling for the multiclass Waldboost.

Weak Learners. The weak learners are a set of M multiclass decision trees with probabilistic outputs, $\mathbf{p}^m(\cdot)$, where $m \in \{1, ..M\}$ is the tree index. The maximum depth of each tree is set to 15, as standardly done in literature. Each decision tree predicts a K dimensional probabilistic output, where K is the number of classes — codebook size. As suggested in [25], we weight the decision boundary in each leaf by the current weights of the training samples falling in that leaf:

$$p_k^m(\mathbf{x}_i) = \frac{\sum_{i \in \mathcal{L}} w_i (y_i^k = 1)}{\sum_{i \in \mathcal{L}} w_i}, \forall k \in \{1, ..K\}, \qquad (1)$$

where $\mathcal{L}$ — the leaf reached by sample $\mathbf{x}_i$, w_i — the weight associated with $\mathbf{x}_i$, and y_i^k the label of $\mathbf{x}_i$ and class k. The weights of the training samples are only used in this step and reset for each weak learner.

Data Pool Sampling. We follow the standard approach and initialize the weights, $\mathbf{w}$, with $\frac{1}{N}$, where N is the number of training samples for the current weak classifier. After training

each weak classifier the weights are updated as follows:

$$w_i = w_i \exp\left(-\frac{K-1}{K}\sum_k^K y_i^k \log p_k^m(\mathbf{x}_i)\right), \quad (2)$$

where the labels for the current sample are:

$$y_i^{k'} = \begin{cases} 1, & \text{if } k' = k, \\ \frac{-1}{K-1}, & \text{otherwise.} \end{cases} \quad (3)$$

During training we sample a 10^{th} of the complete number of training patches to be used for learning each weak classifier. Following [30], we use the QWS+trimming (Quasi-random Weighted Sampling with trimming) for this step. Although each weak classifier is trained on a subset of the training data, the weight update (eq. 2) is applied over the predictions of the weak classifier on the complete data.

Strong Classifier. Following [25], the probabilities of each weak classifier are transformed into real-valued scores:

$$s_k^m(\mathbf{x}_i) = (K-1)\left(\log p_k^m(\mathbf{x}_i) - \frac{1}{K}\sum_{k'}^K \log p_{k'}^m(\mathbf{x}_i)\right). \quad (4)$$

The final probabilistic prediction is obtained by taking the softmax over the sum of the scores of the weak classifiers:

$$s_k(\mathbf{x}_i) = \sum_m^M s_k^m(\mathbf{x}_i), \quad (5)$$

$$p_k(\mathbf{x}_i) = \frac{\exp(s_k(\mathbf{x}_i))}{\sum_{k'}^K \exp(s_{k'}(\mathbf{x}_i))}. \quad (6)$$

Discrimiantive Multiclass Waldboost. Rather than testing all the weak classifiers to reach a decision, we can determine if the strong classifier is sufficiently confident in its prediction and stop without evaluating the subsequent weak classifiers. Waldboost [19] selects a stopping threshold for each weak classifier. These thresholds are learned over the strong classifier scores up to the current iteration.

We propose an intuitive deterministic approach to learning stopping thresholds. After training each weak classifier on its current data pool, we employ it together with all the preceding weak classifiers to obtain a strong prediction on a validation set, denoted by $\bar{\mathbf{x}}$. The strong classifier scores (eq. 5) up to the current step on the validation set are input to a new decision tree. We train one such stopping decision tree per weak classifier, m. The stopping classifiers return K class probabilities. We choose decision trees as stopping classifiers for both consistency as well as efficiency.

At test time, the strong classifier scores up to the current step, m, are passed on to the current decision tree used for stopping. The stopping classifier decides whether to stop or continue with the estimation of the next boosting weak classifier. This is done by evaluating the output of the stopping classifier against a fixed desired class probability, α:

$$\max_k \left(\text{Stop}_k^{M'}\left(\sum_{m=1}^{M'\leq M} s_k^m(\bar{\mathbf{x}}_i)\right)\right) \geq \alpha, \quad (7)$$

where $\text{Stop}_k^{M'}(\cdot)$ — stopping probability for class k at M'.

	HOF	Adaboost HOG/HOF	Waldboost HOG/HOF	$3D$ HOF	Waldboost $3D$ HOF
Time/Frame	0.60 s	4.00 s	0.60 s	7.50 s	0.60 s

Table 1. Run-times of the proposed featureless method versus descriptor extraction. The proposed method is on par with optimized HOF descriptor estimation and ten fold faster than $3D$ HOF.

3.2. Computational Requirements

The speed of dense descriptor extraction has improved considerably as most of the available implementations rely on integral images [31]. However, when the temporal dimension is used — $3D$ descriptors, or motion information — flow-based descriptors, the computational requirements increase. Table 1 shows the times for the computation of different descriptors as well as the Adaboost/Waldboost run-times for predicting on the patches of a complete frame. The runtime numbers are obtained running a single-thread, unoptimized implementation in C++. The Waldboost prediction is as fast as the HOF descriptor extraction over the same frame. However, when predicting a codebook assignment based on $3D$ motion features, boosting approaches are faster while Waldboost is one order of magnitude faster.

4. EXPERIMENTS

We compare against the representations we learn from — BOW (bag-of-words) with k-means codebooks. The codebooks are extracted over HOF, HOG and $3D$ HOF descriptors and used to define the multiclass Waldboost labels. More powerful representation such as Fisher encodings or just the first fully connected layer in a pretrained CNN could also be used, by applying a discretization step. However, the aim is to keep the analysis simple and test the feasibility of transitioning in one step from the low-level features to the final video representation. Therefore, in our experimental setup we use small codebooks and input grayscale patches.

4.1. Experimental Setup

Experiment 1 employs codebooks of only 100 dimensions obtained by applying k-means over 100,000 descriptors and compare against the standard BOW baseline. For the $3D$ descriptors we use larger codebooks — 1000 D — which are also obtained by employing k-means over a set of 100,000 training descriptors. The $3D$ motion descriptors are computed over 8 pairs of frames. **Experiment 2** discards the codebook altogether from both training and test and consider

	HOG		HOF		3D HOF		
	BOW	Waldboost	BOW	Waldboost	BOW	Waldboost	BOW & Waldboost
MAP	**44**%	41%	**37**%	32%	45%	36%	**50**%

Table 2. MAP (mean average precision) scores on UCF11 when using the standard BOW video representation as compared to the representation obtained from the proposed multiclass Waldboost predictions. The results of the featureless approach are comparable with the baseline, while the combination of the two gains in performance.

	Linear SVM	Adaboost	Waldboost
MAP	16%	41%	**41**%
Time/frame	15.00 sec	4.00 sec	0.60 sec

Table 3. Performance of different learning algorithms when learning the mapping from input grayscale values to the HOG codebook assignment. The proposed multiclass Waldboost method achieves better performance than linear SVM and at the same time gains in efficiency over Adaboost at no loss in performance.

each descriptor to be its own cluster center. All experiments use 1000 weak classifiers, each trained on 24 randomly sampled data dimensions. The patch sizes for descriptor extraction as well as for boosting are of 24×24 px. For the multiclass Waldboost we set the stopping threshold, α, to .97 as this proved effective in practice. All experiments report MAP scores (Mean Average Precision) on UCF11 [32] action recognition dataset.

4.2. Experiment 1: Featureless

4.2.1. Experiment 1.1: Waldboost vs. Other Algorithms

Table 3 displays comparative results when learning the mapping from grayscale input values to the HOG codebook assignment. Waldboost manages to outperform by a large margin the linear SVM classifier as it focuses the learning on the informative data dimensions. At the same time, the proposed multiclass Waldboost brings gain in efficiency at no loss in performance when compared with Adaboost, although it analyzed only a subset of the weak classifiers.

4.2.2. Experiment 1.2: Learning vs. Feature Extraction

This experiment tests the feasibility of the featureless aim. Given a set of grayscale input patches together with their codebook assignments over the associated descriptors, it trains a multiclass Waldboost. At test time no descriptors or codebooks are used, thus, obtaining a featureless representation. Table 2 depicts the results obtained by the proposed method when compared to the classic BOW approach. The performance of BOW is slightly better than Walboost. The work of [33] reports an accuracy of 55.46% on BOW with SVM and SIFT descriptors on a codebook of 500 dimensions. Our methods based on BOW and Walboost over HOG features using a codebook of only 100 dimensions obtains a competitive accuracy of 43.18% which corresponds to a mean

	BOW	Codebookless	
		Adaboost	Waldboost
MAP	44%	41%	37%

Table 4. MAP scores on UCF11 for the BOW baseline on a 100D codebook as well codebookless Adaboost and Waldboost — no codebooks are used during training. The learned mapping predicts patch IDs rather than codebook IDs. The boosting methods manage to achieve comparable performance to the baseline despite discarding both descriptors and codebooks.

average precision of 41%, as listed in table 2. However, when combining the two representation — as in the case of 3D HOF, the combined representation exceeds in performance both BOW and Waldboost. This indicates that despite using the same starting point — the same codebook and descriptor assignment, the BOW and Waldboost representations encode complementary information.

4.3. Experiment 2: Featureless and Codebookless

Table 4 displays the action recognition performance when the boosting techniques learn from both a featureless as well as codebookless representation. Each patch is considered to be the center of its own cluster, thus the mapping learns to predict patch IDs rahter than codebook IDs. Out of the 100,000 patches considered, only ≈100 unique patch IDs are begin predicted, the rest having zero predictions. Both Adaboost as well as the proposed multiclass Waldboost still manage to learn the underlying structure in the data, despite not making use of either descriptors or codebooks at test-time.

5. CONCLUSIONS

This work analyzes whether we can bypass feature extraction and still attain comparable performance with the framework we learn from. In search for the simplest possible method, we learn a mapping from grayscale values to existing representations such as codebooks. A straightforward multiclass extension of Waldboost is brought forth for learning this mapping. The efficiency as well as the performance of the proposed method are tested in the context of action recognition. Moreover, we also consider video representations based on motion features, as well as discarding the codebook altogether and learning both a featureless and codebookless mapping.

Acknowledgments This research is supported by the Dutch national program COMMIT.

6. REFERENCES

[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003. 1

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005. 1, 2

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004. 1, 2

[4] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *PAMI*, 2010. 1, 2

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012. 1, 2

[6] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, 2015. 1, 2

[7] C. Vondrick, A. Khosla, H. Pirsiavash, T. Malisiewicz, and A. Torralba, "Visualizing object detection features," *CoRR*, 2015. 1, 2

[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. 1

[9] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *ICCV*, 2015. 1, 2

[10] P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Learning to track for spatio-temporal action localization," in *CVPR*, 2015. 1

[11] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014. 1, 2

[12] M. Jain, J. van Gemert, and C. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?," in *CVPR*, 2015. 1, 2

[13] Z. Xu, Y. Yang, and A. Hauptmann, "A discriminative cnn video representation for event detection," *CoRR*, 2014. 1, 2

[14] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013. 1, 2

[15] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *ECCV*. 2014. 1, 2

[16] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *CVPR*, 2015. 1, 2

[17] D. Hall and P. Perona, "Online, real-time tracking using a category-to-individual detector," in *ECCV*. 2014. 1

[18] D. Hall and P. Perona, "From categories to individuals in real timea unified boosting approach," 2014. 1

[19] J Sochman and J. Matas, "Waldboost-learning for time constrained sequential detection," in *CVPR*, 2005. 1, 2, 3

[20] C. Sun and R. Nevatia, "Discover: Discovering important segments for classification of video events and recounting," in *CVPR*, 2014. 1

[21] K. Soomro, A. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, 2012. 1

[22] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *PAMI*, 2012. 2

[23] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *IJCV*, 2013. 2

[24] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *PAMI*, 2010. 2

[25] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class adaboost," *Statistics and its Interface*, 2009. 2, 3

[26] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding and classification for action recognition," in *CVPR*, 2014. 2

[27] D. Oneata, J. Verbeek, and C. Schmid, "Efficient action localization with approximately normalized fisher vectors," in *CVPR*, 2014. 2

[28] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with r* cnn," *CoRR*, 2015. 2

[29] L. Sun, K. Jia, D. Yeung, and B. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *ICCV*, 2015. 2

[30] Z. Kalal, J. Matas, and K. Mikolajczyk, "Weighted sampling for large-scale boosting," 2008. 3

[31] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008. 3

[32] J. Liu, Y. Yang, and M. Shah, "Learning semantic visual vocabularies using diffusion distance," in *CVPR*, 2009. 4

[33] K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, 2013. 4