Fast saliency based pooling of Fisher encoded dense trajectories

Svebor Karaman, Lorenzo Seidenari, Alberto Del Bimbo
{svebor.karaman, {lorenzo.seidenari, alberto.delbimbo} @ unifi.it

University of Florence

Abstract. In this submission we exploit BING computed objectness windows and stabilized optical flow to extract a weight map. We then compute weighted Fisher vectors computing weights according to these maps. We eventually end up with 4x2+1 channels being HOG, MBHx, MBHy and HOF Fisher Vectors computed on all the frame unweighted and applying the weights. Plus we add a linear kernel obtained computing Decaf features on the key frame. For detection we apply a sliding window approach with a window size of 200 frames with 50

Keywords:

1 Introduction

Human action recognition in videos is a popular research topic in computer vision that can lead to mant applications in surveillance and retrieval.Almost perfect performance is achievable controlled conditions lab video datasets [8], [2], recognizing human action in realistic videos such as sports videos is still challenging due to camera and object motion, background distraction. Some of this issues can be partially solved applying modern space-time features like space-time interest points [5] and dense trajectories [9]. Local features are then typically pooled to obtain a fixed sized representation for each video. The Fisher encoding method [7] has proven to be very powerful when applied on images and has recently successfully been applied in videos. All top performers [3], [11], [6] of previous THUMOS Action Recognition challenge used Fisher encoding.

2 Saliency based pooling

We generate a pooling weight map with a techique inspired by [4]. For each frame we first compute an objectness based saliency map R with the fast BING window proposal accumulating all the rectangles in a map. We then compute the motion magnitude map M. We iteratively compute: $w = \overline{M} \odot \overline{R}$ where $\overline{X}$ is a thresholded version of map X. We then set $M = w \odot R$. This iteration is repeated a few times (3 in our setup). An example on four frame of an *IceDancing* sequence is shown in Figure 1.

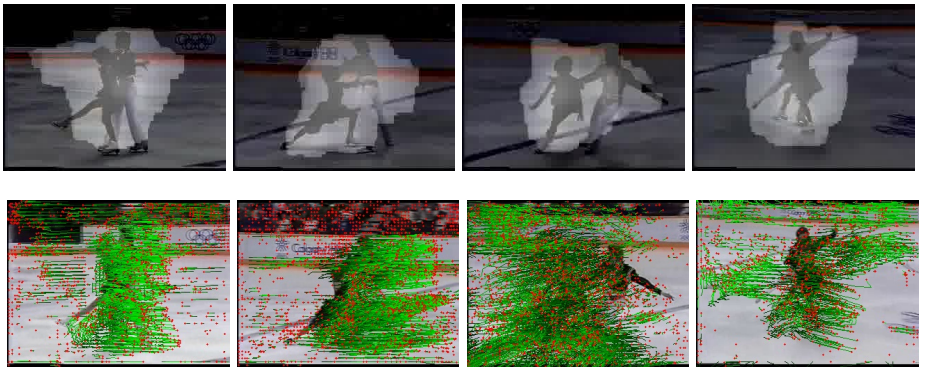

Fig. 1. Fast saliency computed on 4 frames from an ice scating sequence.

3 Features

We employ improved dense trajectories [10] extracted features: HOG, HOF, MBHx, MBHy and TR encoded using Fisher vectors. Fisher vectors are computed on the whole feature set and using the saliency map with soft-fisher encoding [4].

We obtain soft-pooling by computing a weight w_m for each feature $x_m \in X$ to encode. Since $u_\lambda = \sum_{i=1}^{K} \omega_i \mathcal{N}(x; \mu_i, \Sigma_i)$ for a video X we compute the mean and covariance components of a Fisher vector as:

$$\mathcal{G}_n^\mu(X) = \frac{1}{\sqrt{\omega_n}} \sum_{m=1}^{M} w_m \gamma_m^{(n)} \left(\frac{x_m - \mu_n}{\sigma_n^2} \right), \tag{1}$$

$$\mathcal{G}_n^\sigma(X) = \frac{1}{\sqrt{2\omega_n}} \sum_{m=1}^{M} w_m \gamma_m^{(n)} \left(\frac{(x_m - \mu_n)^2}{\sigma_n^2} - 1 \right), \tag{2}$$

where

$$\gamma_m^{(n)} = \frac{\omega_n p_n(x_m)}{\sum_{j=1}^{N} \omega_j p_j(x_m)}. \tag{3}$$

GMM codebooks are trained with 512 Gaussians and all the features are reduced in dimensionality with PCA to 64 elements except TR that are reduced to 20. Moreover we use spatial feature augmentation: each descriptor is first project to the lower dimensional space and then is augmented with x, y and t coordinates normalized with respect to the processed sliding window.

To represent action scene context we apply Decaf features[1] computed, from the last layer on the network pre-trained on ImageNet, on the central crop of the image.

We trained separed linear SVM with C=1 for each feature channel using late fusion, i.e. summing classifiers scores.

4 Detection Strategy

4.1 Classification task

We run our classifiers trained on the trimmed training examples on sliding windows of 200 frames with an overlap of 50%. To select the most likely window containing a given action we first compute all single feature classifier scores $s_f(w)$ for every window w in a video. Than we apply a set of increasing thresholds $t_f = \{\min_w s_f \dots \max_w s_f\}$. We then accumulate all feature windows thresholded scores and take the max of windows. In our preliminary experiments this strategy improved the discriminative power of single feature classifiers.

4.2 Detection task

For detection, we add to the fullScore matrix that we get at in 4.1 the sum of the scores of all temporal window for each class. With this step we try to boost the scores of temporal window for an action that seems to appear all over the video.

Then to finally get the detection we get the max for each temporal window using all classes, and then subsample the matrix for the classes of interest for the detection task.

Then we get the median score value from all classes, and keep detection only if they are both a global max and with a score higher than median value.

5 Experiments

We performed a set of experiments on the 3 training splits of previous year that show how all the different channels are complementary except for the TR descriptor that worsen the result a bit. For this reason we removed the TR descriptor in the final experiments. This results are shown in Table 1.

Features	Full	FG	xyt	no-aug	Average	Split 1	Split 2	Split 3
HOG HOF MBHx MBHy TR	✓	✓	✓	✓	0,8329	0,8228	0,8463	0,8296
HOG HOF MBHx MBHy	✓	✓	✓	✓	**0,8412**	**0,8329**	**0,8551**	**0,8357**
HOG HOF MBHx MBHy	✓	✓	-	✓	0,8337	0,8250	0,8461	0,8301
HOG HOF MBHx MBHy	✓	✓	✓	-	0,8365	0,8270	0,8514	0,8310
HOG HOF MBHx MBHy	-	✓	✓	-	0,8386	0,8263	0,8553	0,8343

Table 1. Preliminary evaluation of feature combinations on cropped videos from Thumo13.

	Run 1	Run 2	Run 3
mAP	0.2919	0.2809	0.2679

Table 2. Results on test set for the three run submitted. Run 1 uses all features comprising DeCaf, FG and xyt-augmented channels, Run 2 uses all features except Decaf, Run 3 uses only FG and xyt-augmented channels without DeCaf. No Run used TR features.

References

1. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. arXiv preprint arXiv:1310.1531 (2013)
2. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. Transactions on Pattern Analysis and Machine Intelligence 29(12), 2247–2253 (December 2007)
3. Karaman, S., Seidenari, L., Bagdanov, A.D., Del Bimbo, A.: L1-regularized logistic regression stacking and transductive crf smoothing for action recognition in video. In: THUMOS'13 Action Recognition Challenge (2013)
4. Karaman, S., Seidenari, L., Ma, S., Del Bimbo, A., Sclaroff, S.: Adaptive structured pooling for action recognition. In: Proc. of British Machine Vision Conference (BMVC) (2014)
5. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Proc of. Computer Vision and Pattern Recognition (CVPR). IEEE (2008)
6. Murthy, O.R., Goecke, R.: Combined ordered and improved trajectories for large scale human action recognition. In: THUMOS'13 Action Recognition Challenge (2013)
7. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Proc of. Computer Vision and Pattern Recognition (CVPR). pp. 1–8. IEEE (2007)
8. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: Proc. of International Conference on Pattern Recognition (ICPR) (2004)
9. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision pp. 1–20 (2013)
10. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: IEEE International Conference on Computer Vision. Sydney, Australia (2013), http://hal.inria.fr/hal-00873267
11. Wang, H., Schmid, C.: Lear-inria submission for the thumos workshop. In: THUMOS'13 Action Recognition Challenge (2013)