LPM for Fast Action Recognition with Large Number of Classes

Feng Shi, Robert Laganière, Emil Petriu
School of Electrical Engineering and Computer Scinece
University of Ottawa, Ottawa, On, Canada
fshi98@gmail.com, {laganier, petriu}@site.uottawa.ca

Haiyu Zhen
Department of Electronics and Information Engineering
Hua Zhong University of Science and Technology, Wuhan, China
zhenhaiyu@mail.hust.edu.cn

1. Introduction

In this paper, we provide an overview of the Local Part Model system for the THUMOS 2013: Action Recognition with a Large Number of Classes[1] evaluations. Our system uses a combination of fast random sampling feature extraction and local part model feature representation.

Over the last decade, the advances in the area of computer vision and pattern recognition have fuelled a large amount of research with great progress in human action recognition. Much of the early progress [1, 5, 14] has been reported on atomic actions with several categories based on staged videos captured under controlled settings, such as KTH [14] and Weizmann [1]. More recently, there are emerging interests for sophisticated algorithms in recognizing actions from realistic video. Such interests involve two prospects: 1) In comparison to image classification evaluating millions of images with over one thousand categories, action recognition is still at its initial stage. It is important to develop reliable, automatic methods which scale to large numbers of action categories captured in realistic settings. 2) With over 100 hours of videos are uploaded to YouTube every minute[2], and millions of surveillance cameras all over the world, the need for efficient recognition of the visual events in the video is crucial for real world applications.

Recent studies [5, 10, 11, 21] have shown that local spatio-temporal features can achieve remarkable performance when represented by popular bag-of-features method. A recent trend is the use of dense sampled points [16, 21] and trajectories [7, 19] to improve the performance. Local Part Model [15] achieved state-of-the-art performance on real-life datasets with high efficiency when combined with random sampling over high density sampling grids. In this paper, we focus on recognize human action "in the wild" with large number of classes. More specifically, we aim to improve the state-of-the-art Local Part Model method on large scale real-life action datasets.

The paper is organized as follows: The next section reviews the LPM algorithm. Section 3 introduces four different descriptors we will use. In section 4, we present some experimental results and analysis. The paper is completed with a brief conclusion. The code for computing random sampling with Local Part Model is available on-line[3].

2. LPM algorithm

Inspired by the *multiscale, deformable part model* [6] for object classification, we proposed a 3D multiscale part model in [16]. However, instead of adopting deformable "parts", we used "parts" with fixed size and location on the purpose of maintaining both structural information and local events ordering for action recognition. As shown in Figure 1, the local part model includes both a coarse primitive level *root* feature covering event-content statistics and higher resolution overlapping *part* filters incorporating local structural and temporal relations.

More recently, we [15] applied random sampling method with local part model over a very dense sampling grid and achieved state-of-the-art performance on realistic large scale datasets with potential for real-time recognition. Under the local part model, a feature consists of a coarse global *root* filter and several fine overlapped *part* filters. The *root* filter is extracted on the video at half the resolution. This way, a high density grid can be defined with far less samples. For every coarse *root* filter, a group of fine *part* filters are computed at full video resolution and at locations relative to their *root* filter reference position. These *part* filters

[1] http://crcv.ucf.edu/ICCV13-Action-Workshop/index.html
[2] http://www.youtube.com/yt/press/statistics.html
[3] https://github.com/fshi/actionMBH

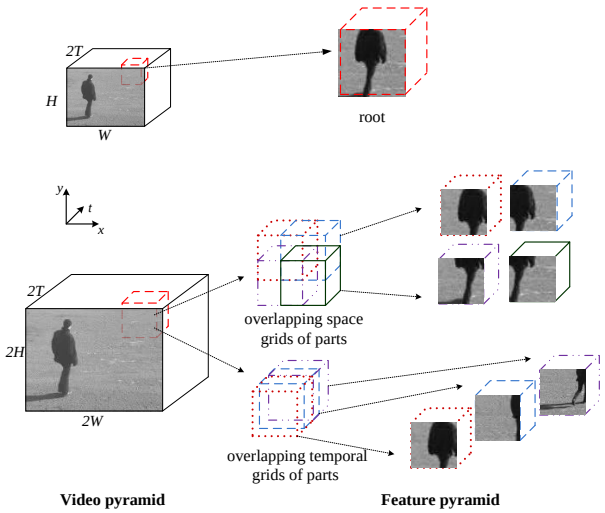

Figure 1: Example of Local Part Model defined with root filter and overlapping grids of part filters.

contain the fine-grained information required for accurate recognition.

To improve the efficiency of LPM computation, two integral videos are computed, one for the *root* filter at half resolution, and another one for the part filters at full resolution. The descriptor of a 3D patch can then be computed very efficiently through 8 additions multiplied by the total number of root and parts. Apart from descriptor quantization and normalization, most cost associated with feature extraction is spent on accessing memory through the integral videos.

Because it uses random sampling, the method does not require feature detection, which greatly improves processing speed. One LPM feature includes a *root* spatial-temporal(ST) patch and a group of ST *part* patches. In our previous paper [15], each of these patches is represent by a local descriptor (*e.g*. MBH, HOG3D, HOF *etc*.) as a histogram, and all the histograms are concatenated into one vector that is 9 (1 *root* + 8 *parts*) times the original feature dimension. Such approach, however, can result in large codeword quantization errors. It also obscures the discriminative power of independent *root* filter and *parts* filters.

In this paper, we make an improvement over our previous work. Instead of simply concatenating 1 *root* descriptor and 8 *parts'* descriptors, we treat the *root* and 8 *parts* as two separate channels. For each channel, a standard bag-of-features approach is applied. The resulting histograms of visual word occurrences from *root* and *parts* are concatenated into one histogram for SVM classification. We will discuss this in details in our experimental section.

3. Local descriptors

For each given sample point (x, y, t, σ, τ), a feature descriptor is computed for a 3D video patch centred at (x, y, t). The descriptors are critical for the performance of the video recognition. In our experiments, four types of local descriptors are computed to encode local motion pattern and structure information.

HOG descriptor was introduced by Dalal and Triggs in[3] for human detection. It is based on the popular SIFT descriptor [12].

HOF descriptor was first used by Laptev *et al*. [11] to combined with HOG to incorporate both local motion and appearance.

HOG3D descriptor was proposed by Kläser [8]. It is built up on 3D oriented gradients. Although it is very efficient to compute 3D gradients, the orientation quantization with polyhedron for each sub-blocks is relatively cost considering a lot of patches sampled.

MBH descriptor was introduced by Dalal *et al*. in[4] and used by Wang *et al*. [19] on action recognition to achieved state-of-the-art performance. The derivatives are computed separately for the horizontal and vertical components of the optical flow, which results in motion compensation. The the horizontal and vertical MBH descriptors can be computed based on the derivatives.

3.1. Parameters

We strictly follow the parameter settings as those in [15]:

HOG3D. The parameters are: number of histogram cells $M = 2$, $N = 2$; number of sub-blocks $1 \times 1 \times 3$; and polyhedron type dodecahedron(12) with full orientation; the cut-off value is $c = 0.25$. The minimal patch size is $16 \times 16 \times 10$. One HOG3D descriptor for a 3D patch has a dimension of 96, our local part model feature has a dimension of 96 for *root* channel and 768 for *parts* channel (8×96).

HOG, HOF and **MBH.** The minimal patch size is $16 \times 16 \times 14$ for HOG and $20 \times 20 \times 14$ for HOF and MBH. Each patch is subdivided into a grid of $2 \times 2 \times 2$. With 8 bins quantization, one descriptor of HOG, HOF , MBHx or MBHy has a dimension of 64 ($2 \times 2 \times 2 \times 8$). Our local part model feature has a dimension of 64 for *root* channel and 512 for *parts* channel (8×64). In addition, we also treat MBHx and MBHy as two separate channels.

4. Experiments

To demonstrate the performance of the improved LPM, we evaluated our method on two large-scale action benchmarks, the UCF101 [17] and the HMDB51 [9] datasets. We strictly follow the experimental settings as those in [15]. For efficiency, we down-sample the UCF101 and HMDB51 videos to half the spatial resolution for all our experiments. We randomly sample 3D patches from the

dense grid, and use them to represent a video with a standard bag-of-features approach. The sampled 3D patches are represented by descriptors, and the descriptors are matched to their nearest visual words with Euclidean distance. The resulting histograms of visual word occurrences are fed into a non-linear SVM implemented by LIBSVM [2] with histogram intersection kernel [18]. For multi-class SVM, we use one-versus-all approach, which is observed [7] to have better results than one-against-one multi-class SVM.

To generate codewords, we randomly selected 120,000 training features, and used k-means to cluster them into 2000 visual words for each channel. For each input video, we set the maximal video length to 160 frames. If the video is larger than 160 frames, we simply divide it into several segments, and select features at same rate for each segment. We randomly sample 10k features. Each feature has 1 root and 8 parts, which are represented by a descriptor as root vector and parts vector, respectively. The feature vectors are matched to nearest visual words with Euclidean distance. After matching features with codewords for the root channel and parts channel separately, we concatenate the resulting histograms of visual word occurrences.

For better performance, multiple descriptors can be combined to provide complementary information. To combine multiple channels of different descriptors, we follow [15] by using histogram intersection kernel weighted with descriptors' discriminative power:

$$K_{IH}(x_i, x_j) = \sum_c \frac{w^c}{max(w^c)} min(x_i^c, x_j^c), \qquad (1)$$

where w^c is classification accuracy for the c-th channel, which can be learnt from the training data. $max(w^c)$ is the maximal value from w^c of all channels.

4.1. Datasets

The **UCF101** dataset [17] is by far the largest human action dataset with 101 classes and 13320 realistic video clips taken from YouTube. All clips have fixed frame rate and resolution of 25 FPS and 320×240 respectively. The clips of one action class are divided into 25 groups which contain 4-7 clips each. The clips in one group may have similar background or be played by the same subjects. The dataset is very large and relatively challenging due to camera motion, cluttered background, large scale variations, etc. We report mean accuracy over three train/test splits as author's website[4]. For split 1, split 2 and split 3, , clips from groups 1-7, groups 8-14 and groups 15-21 are selected respectively as test samples, and the rest for training.

The **HMDB51** dataset [9] contains 51 action categories, with at least 101 clips for each category. It is perhaps the most realistic and challenging dataset. The dataset includes a total of 6,766 video clips extracted from Movies, the Prelinger archive, Internet, Youtube and Google videos. Three distinct training and testing splits have been selected from the dataset, with 70 training and 30 testing clips for each category. We used the original non-stabilized videos with the same three train-test splits as the authors [9], and report the mean accuracy over the three splits in all experiments.

[4] http://crcv.ucf.edu/data/UCF101/UCF101TrainTestSplits-RecognitionTask.zip

Method		HMDB51	UCF101
HMDB51 [9]		23.2%	–
ActionBank [13]		26.9%	–
Random sampling [15]		47.6%*	–
Dense trajectories [20]		46.6%*	–
DCS descriptor [7]		52.1%*	–
UCF101 [17]		–	43.9%
Ours	HOG	23.7%	48.7%
	HOF	35.3%	59.6%
	HOG3D	36.0%	59.0%
	MBH	50.6%	76.3%
	Combined	**55.2%***	**78.9%***

Table 1: Comparison of average accuracy on UCF101 and HMDB51 with state-of-the-art methods in the literature. Those marked with * are results with combined descriptors.

4.2. Results

Table 1 shows the results and comparison of our method on **HMDB51** and **UCF101** with the state-of-the-art. For efficiency, we use 2000 codewords for root channel and parts channel on all descriptors. For multiple descriptors (as shown in Table 1), we combined all 4 channels with Equation 1. On **HMDB51**, the results show consistently good performance on all descriptors. Our method outperforms all previous results in the literature for both single descriptor and combined descriptors. In particular, we obtained 50.0% average accuracy with MBH descriptor, which is 6.7% more than the best single descriptor approach [20]. With multi-channel approaches, our method shows a performance of 55.2%, which exceeds the best reported results in [7] (52.1%).

On **UCF101**, we are the first to report the evaluation, and our results are obtained with three distinct training and testing splits as described in Section 4.1. We achieved 76.3% with MBH descriptor and 78.9% with four descriptors. The confusion matrix for all 101 actions is shown in Figure 2. Our results significantly outperform the baseline results (43.9%) of the UCF101 [7].

Dataset	BoF matching method	Speed (frames per second)				Mean accuracy
		Integral video	Sampling	BoF matching	Total fps	
HMDB51	FLANN	36.8	114.1	118.2	23.1	48.2%
	Brute Force	34.2	115.9	15.6	9.8	50.6%
UCF101	FLANN	51.2	170.0	185.9	32.5	75.7%
	Brute Force	45.4	167.4	22.6	13.8	76.3%

Table 2: Average computation speed with single core at different stages in frames per second. The MBH descriptor is used with 2000 words per channel, and 10K features are sampled in the experiment. The optical flow computation is included in "Integral video".

4.3. Computational efficiency

Table 2 summarizes the efficiency comparison at different stages for HMDB51 and UCF101 dataset when using MBH descriptor. Brute Force and FLANN methods are evaluated and compared. Except for codewords matching, all other stages are same. There are small speed differences between HMDB51 and UCF101. One explanation is that HMDB51 videos have variable resolution. Also, it has fewer frames per video which results in overhead for the computation. The computation time was estimated on an Intel i7-3770K PC. We did not parallelize our code and only used a single core of the CPU.

Note that for computational efficiency, we down-sample the UCF101 and HMDB51 videos to half the spatial resolution for all our experiments. We also use 2000 codewords for all our experiments. We observed improved performance for standard 4000 codewords or by using full spatial resolution with parameter tuning.

5. Conclusions

This paper introduced multi-channel approach for LPM algorithm for efficient action recognition. Compared to the standard representation, the multi-channel approach improves the performance with less codewords. Our results achieved state-of-the-art on two realistic large scale datasets, UCF101 and HMDB51.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes, 2005. 1

[2] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011. 3

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE, 2005. 2

[4] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages 428–441. Springer, 2006. 2

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. 2nd Joint IEEE Int Visual Surveillance and Performance Evaluation of Tracking and Surveillance Workshop*, pages 65–72, 2005. 1

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8, 2008. 1

[7] M. Jain, H. Jégou, P. Bouthemy, et al. Better exploiting motion for better action recognition. In *CVPR*, 2013. 1, 3

[8] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, pages 995–1004, 2008. 2

[9] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011. 2, 3

[10] I. Laptev and T. Lindeberg. Space-time interest points. In *Proc. Ninth IEEE Int Computer Vision Conf*, pages 432–439, 2003. 1

[11] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008. 1, 2

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, pages 91–111, 2003. 2

[13] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, pages 1234–1241, 2012. 3

[14] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR (3)*, pages 32–36, 2004. 1

[15] F. Shi, E. Petriu, and R. Laganiere. Sampling strategies for real-time action recognition. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition. IEEE*, 2013. 1, 2, 3

[16] F. Shi, E. M. Petriu, and A. Cordeiro. Human action recognition from local part model. In *Proc. IEEE Int Haptic Audio Visual Environments and Games (HAVE) Workshop*, pages 35–38, 2011. 1

[17] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. Technical Report CRCV-TR-12-01, CRCV, University of Central Florida, 2012. 2, 3

[18] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991. 3

[19] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011. 1, 2

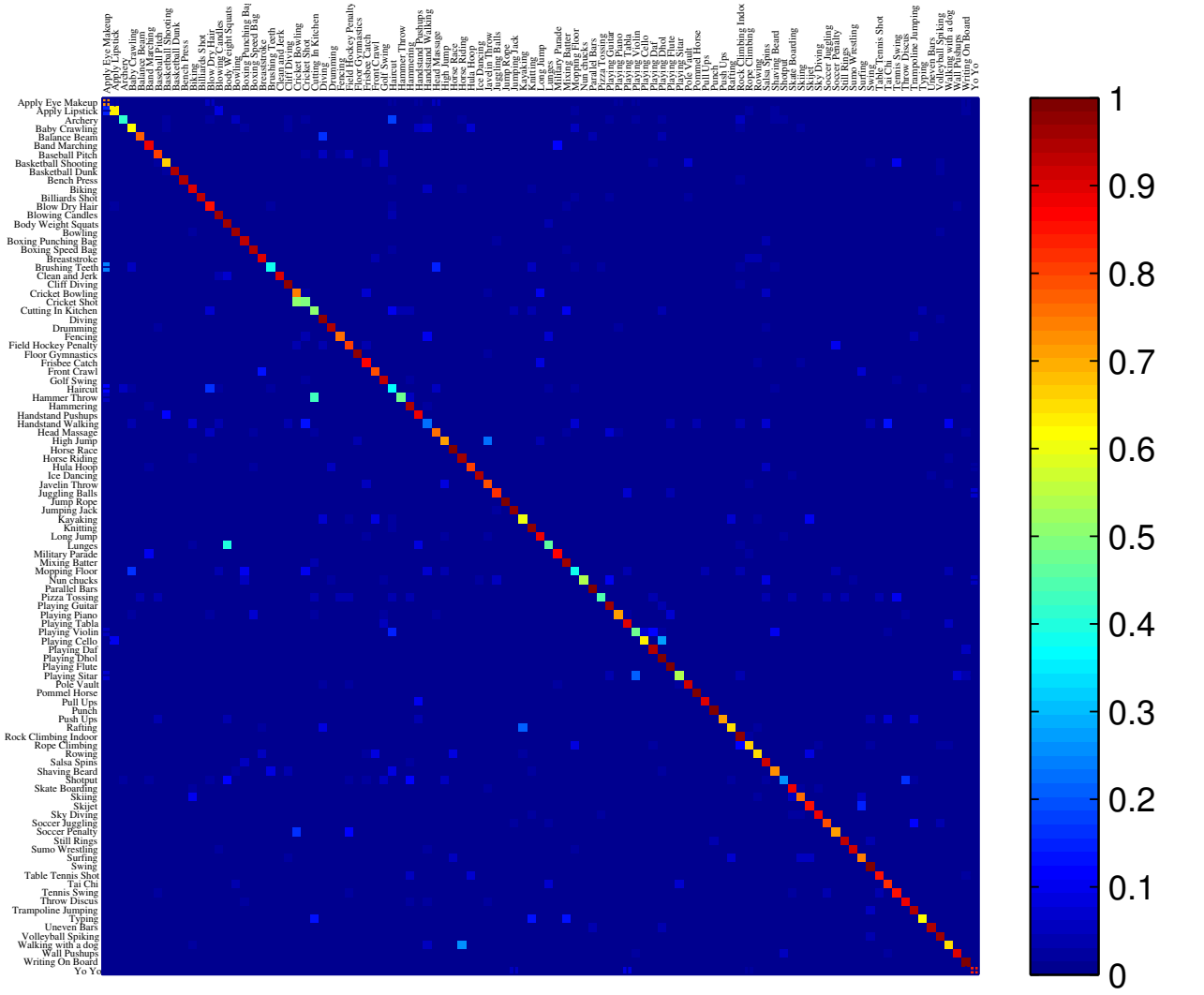

Figure 2: Confusion matrix for UCF101 dataset using 4 descriptors combined and PCA96 BoF matching. The test results of training/testing split 3 (Section 4.1) are used to generate the matrix.

[20] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. Technical report, INRIA, 2012. 3

[21] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, pages 127–137, 2009. 1