Non-Intrusive Human Activity Monitoring in a Smart Home Environment

S. Mohsen Amiri
Dept. of Electrical & Computer Eng.
University of British Columbia
Canada
mohsena@ece.ubc.ca

Mahsa T. Pourazad
TELUS Communications Inc. &
University of British Columbia
Canada
pourazad@ece.ubc.ca

Panos Nasiopoulos, Victor C.M. Leung
Dept. of Electrical & Computer Eng.
University of British Columbia
Canada
{panos, vleung}@ece.ubc.ca

***Abstract*— Non-intrusive activity monitoring of occupants in a home environment plays an important role in developing the next generation of smart environments and remote health monitoring systems. One important challenge in this research area is lack of a comprehensive dataset. In this paper, we introduce a new dataset related to a smart home environment, which can be used for human activity recognition. In addition, we benchmark our proposed human action recognition algorithm and some other state-of-the-art methods using our dataset.**

Keywords—smart home and human action recognition

I. INTRODUCTION

There is an increasing need for monitoring people's health condition and functional status within their personal home environment. Such activity is very important today due to the increasing cost and the discomfort/intrusiveness of methods presently used at caregiving facilities as well as the accelerated rate of aging population around the world [1]. For the above reasons, in the past few years many systems have been developed for health-care monitoring applications including wearable sensors and vision-based systems [2]. For non-intrusive occupant monitoring in smart environments, vision-based systems are preferred over wearable devices, since high volume of information can be extracted from them without intrusion into occupants' privacy. Automatic recognition of human activities from the extracted information plays a crucial role in non-intrusive occupant monitoring by vision-based systems.

In the past few years, many researchers in academia and industry have focused on the problem of human action recognition for different applications. The crucial and challenging issue here is having access to benchmark datasets for specific applications in order to effectively train and test proposed techniques. Although simple and general video datasets such as KTH [3] and IXMAS [4] are available to researchers for evaluating human action recognition algorithms, the algorithms with high performance on these datasets may not necessarily achieve the same performance when they are applied to more complex datasets for different applications. In general, due to the natural complexity of analyzing human actions in video, developing a highly tuned algorithm with a high accuracy on one dataset for a specific application cannot guarantee that the same algorithm achieves even acceptable performance on other datasets which have been captured with different applications in mind. Thus, it is of high importance to generate more complete and task specific datasets for different applications. For example, Hollywood's dataset specifically has been designed for event/action detection in movie applications [5], while the TRECVIDMED dataset focuses on event detection for Internet videos (such as YouTube) applications [6].

To the best of our knowledge there is no benchmark dataset for occupant monitoring in home environments. Most of the available action recognition datasets do not provide realistic examples of human action in a home environment. The availability of dedicated datasets for home occupant monitoring, which represent the occupants' natural behaviour/action in different events at home, is a key factor in developing effective human-action recognition algorithms for smart home applications.

The focus of our work is to provide the industry and academia communities with an action recognition dataset consisting of realistic clips for non-intrusive occupant monitoring in smart environments. To this end, our group at the digital Multimedia Lab (DML) of the University of British Columbia (UBC) created a new dataset called "*DMLSmartActions"* specifically for monitoring occupants in a home environment and made this dataset publicly available to the research community. In collecting this dataset, the objective was to capture realistic behaviour/action of occupants in the home environment. Our dataset will help researchers to develop more effective human action recognition algorithms that can be used in smart home environments and caregiving facilities. In our study, we also investigate the performance of the latest human action recognition algorithms [7, 8, 9] over our dataset. Note that these algorithms have been developed based on available simple and general video datasets such as KTH [3] and IXMAS [4]. Using the *DMLSmartActions* dataset, we also study the effect that different settings of these algorithms have on action recognition accuracy..

The rest of this paper is organized as follows: Section II provides a short background on the latest human action recognition algorithms, Section III provides information about our *DMLSmartActions* dataset that is specific to human actions in a home environment, Section IV discusses our experiments

that evaluate the performance of the latest human action recognition techniques on the *DMLSmartActions* dataset, and conclusion is drawn in Section V.

II. Overview of Human Action Recognition Algorithms

During the past few years, several different human action recognition algorithms have been proposed. Many of these algorithms are based on analyzing human silhouettes, treating them as three-dimensional (3D) objects (space-time) and using 3D object recognition techniques to classify human actions [10], [11]. Due to occlusion and cluttering issues in the background, tracking the human body is a challenging and error-prone task. Thus, the applicability of these techniques for human action recognition in real scenes is limited [12].

Recent studies have shown that local spatiotemporal patches of video signals carry suitable information for human action recognition [7, 8, 9, 14]. In other words, the features extracted from spatiotemporal patches can be used for human action recognition. The features extracted from spatiotemporal patches can be well described by a model called "Bag-of-Word (BoW)" [13]. BoW treats a video sequence as a set of unordered appearance of spatiotemporal features, quantizes them into spatiotemporal words and computes a compact histogram representation, which is used as the global representation of the video stream. To create spatiotemporal visual words (high-level features), usually an unsupervised clustering algorithm (e.g., k-means) is applied to the extracted features [7]. However, recent studies show that by replacing unsupervised clustering with a sparse coding (SC) algorithm, recognition algorithms can achieve higher recognition accuracies in many computer vision tasks [14]. The main idea in SC is that any high dimensional signals, such as speech signals, natural images and video sequences, can be approximated by using a linear combination of a small number of basis (words), where these basis (words) are chosen from an appropriate collection of basis (group of words/dictionary). In our application, assume that x_i is a detected spatiotemporal feature ($x_i \in R^m$) which can be approximated by $D\alpha_i$ if $\|\alpha_i\|_0 \ll m$ (i.e., number of non-zero components of α_i is small), where D is a dictionary ($D \in R^{m \times p}$), α_i is a sparse representation of ($\alpha_i \in R^p$). Given a dictionary D, the sparse representation of x_i can be found by minimizing the reconstruction error of x_i plus a penalty term as follows:

$$l(\alpha_i; x_i, D) = \frac{1}{2}\| x_i - D\alpha_i\|_2^2 - \lambda\|\alpha_i\|_1 \quad (1)$$

where $\lambda\|\alpha_i\|_1$ is a penalty term (λ controls the sparsity). Minimizing $l(\alpha_i; x_i, D)$ for a given dictionary D is a convex problem. A wide range of efficient algorithms has been proposed to solve this problem [15].

In human action recognition applications, once we have clustered spatiotemporal words (obtained by k-means or SC), which describe local patches of the video sequence in time and spatial domain, we need to recognize the action within the video clip. Basically, we require an algorithm that aggregates the information in different spatiotemporal patches, and build a global descriptor of the whole video clip, which represents the action within the clip. Several different aggregation techniques called "pooling" techniques have been proposed, such as average-pooling and max-pooling. Among the different pooling techniques, algorithms with max-pooling achieve the highest accuracy [8], [9], [16]. Having a group of words, max-pooling is choosing the element-wise maximum of the absolute value of these words. Assume that we have *n* visual words in a video clip, i.e., $A = [\alpha_1, \alpha_2, \cdots, \alpha_n]$. If α_i^k represents the k^{th} element of the i^{th} word, then the k^{th} element of the result of max-pooling can be defined as follows:

$$z^k = max\{|\alpha_1^k|, |\alpha_2^k|, \cdots, |\alpha_n^k|\} \quad (3)$$

where z is a global descriptor for the video clip. Once the global descriptor z of the whole video clip is obtained, a classifier such as a Support Vector Machine (SVM) with different kernels (such as χ^2 kernel [7], intersection kernel, and linear kernel [8], [9]) can be used for action classification.

As it can be observed from (3), max-pooling uses the absolute value of the components and ignores their sign. This may result in loosing valuable information, which consequently degrades the accuracy of the action-classifier [9]. To address this problem, our algorithm proposed in [9] applies a non-negative sparse coding (NNSC) algorithm, which avoids generating negative words by adding positivity constraints on the elements of each word within the video as follows:

$$l_{NN}(\alpha_i; x_i, D) = \frac{1}{2}\| x_i - D\alpha_i\|_2^2 - \lambda\|\alpha_i\|_1 \quad (4)$$
$$\text{subject to: } \forall i, \alpha_i^k \geq 0$$

Experimental results show that non-negative sparse coding provides better visual words for the human action recognition task and can achieve high accuracies for some datasets such as KTH and IXMAS.

III. *DMLSmartAction*: Dataset for Human Action in Smart Environments

To demonstrate the real situation in a home environment, we simulate the living room environment by building two separate living rooms in our Digital Multimedia Lab (DML) and capture our own dataset ("*DMLSmartActions*"). In our setting we capture the video clips using static cameras (2 High-definition RGB cameras) and one Kinect sensor. In total, the dataset has two HD RGB streams from the two HD cameras plus one VGA RGB stream and one VGA depth stream from the Kinect sensor. To add more variation to the captured data, we did not fix the location and orientation of the cameras, although the cameras were static. Due to using static cameras, this dataset does not require handling camera motion or zoom changes. Before labeling the dataset, we changed the frame rate of all the video streams to 30 fps using FFMPEG and manually synchronized them. Figure 1 shows a sample of our camera arrangement for recoding this data. Note that the Kinect sensor was always located between the two HD cameras in different scenes.

Fig 1. An example of the camera setting used for capturing the *DML-SmartAction dataset.*

Sixteen subjects helped us during recording (6 females and 12 males). During data collection, we asked the different subjects to perform a series of actions with their own natural style, i.e., we did not provide them with any instructions about how or when to perform these actions. This allowed the subjects to act naturally. The reason is that when people act naturally, there is considerable temporal correlation between different actions. For example, observing a "writing" action or "reading" after "sitting-down" may be considered a natural activity.

Once the data was collected, we manually classified/labelled it into twelve different actions, which are common in people's day-to-day life in a home environment. The *DMLSmartActions* dataset currently contains 932 annotated video samples, which represent the following twelve different actions: clean-table, drink, drop-and-pickup, fell-down, pick-something, put-something, read, sit-down, stand-up, use-cellphone, walk, and write. Figure 2 shows sample frames from the *DMLSmartAction* dataset. The wide range of the actions and the large number of subjects with different genders in this dataset make it a unique resource for research in the field of human action analysis in home environment. This dataset is publicly available to the research communities [22].

IV. Implementation and Evaluation

In our study, we investigate the performance of current action recognition algorithms on the *DMLSmartActions* dataset. The following sub-sections elaborate on the implementation aspects of our experiments (adjusting the parameters of each algorithm) and provide a comparison between the performance of these different algorithms using our collected dataset.

A. Implementation and parameters setting

In our experiment, we adopt the suggested framework by [7] for human action recognition, which includes spatiotemporal feature extraction, building visual words, max-pooling and classification. To extract spatiotemporal features, we use the Hariss3D algorithm [13] as the feature detector and STIP as feature descriptors [18]. In order to speed up the feature extraction process, we down-sampled the HD resolution videos provided in the *DMLSmartActions* dataset to 480×270 pixels. Note that in our experiment we did not use the Kinect sensor data (depth and RGB signal), which is available in the dataset. The reason for that none of the existing algorithms is designed to take advantage of such information.

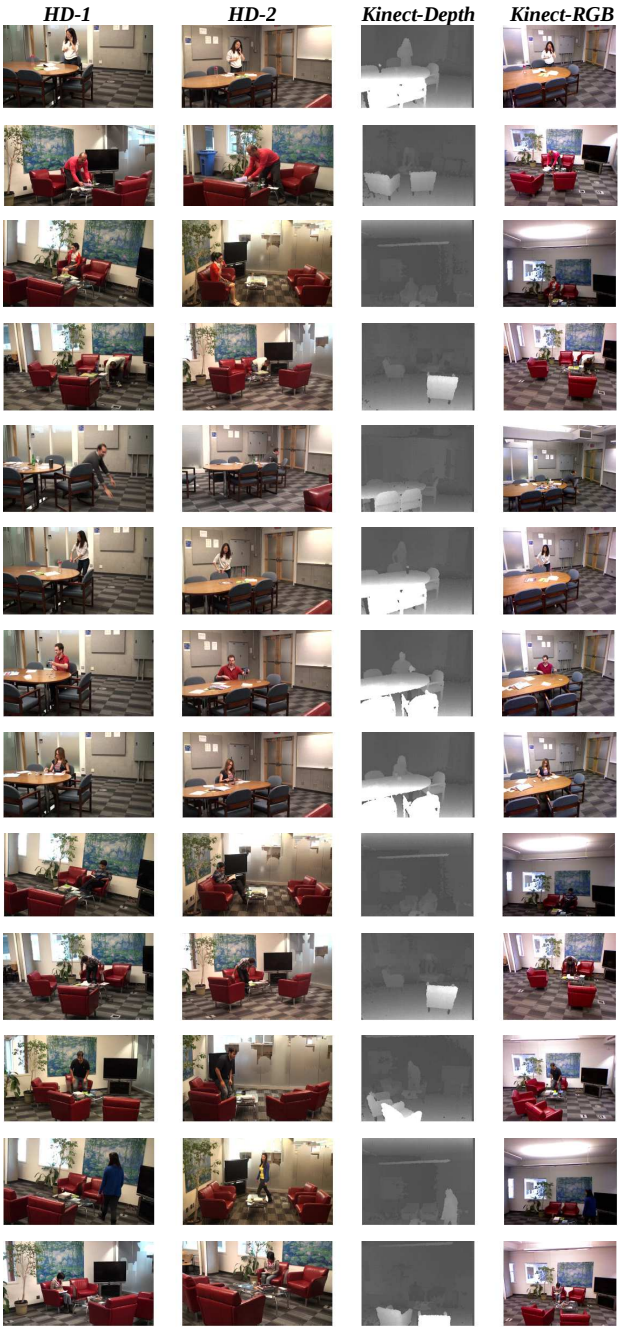

Fig 2. Sample frames from DML-SmartActions dataset. HD-1 and HD-2 are sample frames from two HD cameras. Kinect-Depth is the VGA depth stream from the Kinect sensor, and Kinect-RGB is the VGA RGB stream from the Kinect sensor.

To construct visual words from spatiotemporal features, k-means, SC and NNSC are used. The SPAMS (SPArse Modeling Software) library is used for SC and NNSC implementations [19]. In this experiment, as suggested by [8], [9], [20], the value of λ (sparsity controller in (1)) is set to $\frac{1.2}{\sqrt{m}}$, where m is the dimension of the detected features by STIP (default m=160). Note that in order to find the most

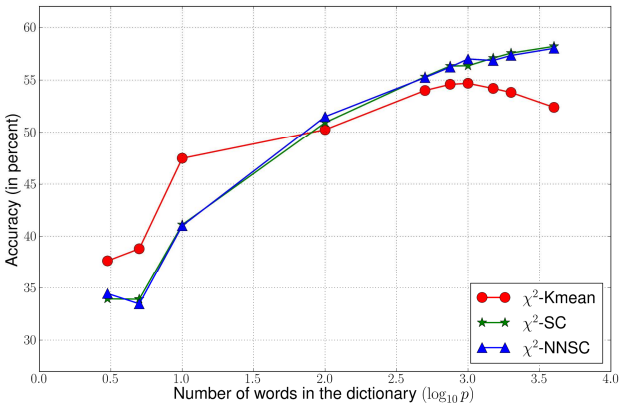

Fig. 3. Recognition accuracy when a χ^2-SVM is used as the classifier.

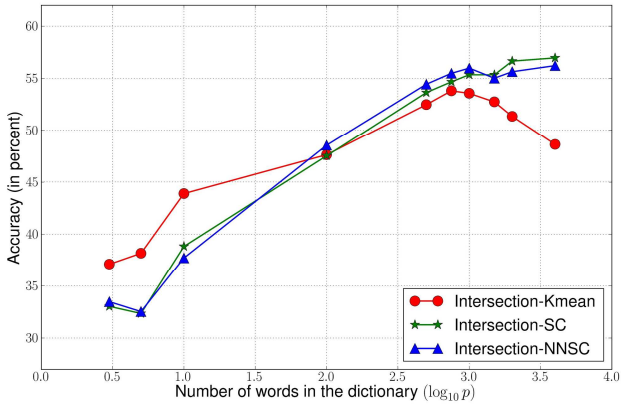

Fig. 4. Recognition accuracy when an Intersection -SVM as the classifier.

discriminative dictionaries, we perform extensive search over dictionaries with different number of words. For implementing k-means clustering, we use the k-means++ algorithm proposed by [21], which is much faster than the regular implementation of k-means.

The classifier used for the action recognition algorithm is the multi-class SVM developed by [22] with three different kernels (χ^2 kernel, intersection kernel, and linear kernel), and with a L_2-regularized and L_2-loss cost function. To find the optimum parameters for SVM, we used a cross-validation schema and searched over a large set of different values to find a setting that generates the highest accuracy.

For evaluating the different algorithms we use the Leave-One-Out (LOO) strategy. In our LOO, we iterate over different subjects and each time exclude all the clips belonging to one subject and train the system using the remaining clips. Then, the excluded clips are used to test the trained system and find its accuracy. Finally, the overall accuracy of each algorithm is calculated as the average of the accuracy values obtained over all iterations.

B. Performance evaluation

Once the human action recognition algorithms are implemented as explained in the previous subsection, we compare their performance on the *DMLSmartActions* dataset. Figure 3 shows the accuracy of the human action recognition algorithms which use k-means, SC, and NNSC to construct words and use a χ^2 –SVM classifier to classify the actions within dataset. The case where k-means is used with χ^2–SVM classifier is the proposed approach by [7]. As it can be observed, the accuracy improves as the number of words in the dictionary increases. We also note that when the number of words is more than 100, the accuracy for SC and NNSC surpasses that of k-means.

TABLE 1. ACHIEVED ACCURACIES WHEN DIFFERENT METHOD ARE USED

Classifier	k-means	SC	NNSC
χ^2- SVM	54.69%	**58.20%**	58.01%
Intersection-SVM	53.81%	**56.96%**	56.22%
Linear SVM	50.47%	56.68%	**57.55%**

Figure 4 shows the accuracy of the human action recognition algorithms, which use k-means, SC, and NNSC to construct words and use an intersection –SVM classifier to classify the actions within dataset. It can be seen that by increasing the number of words in the dictionary, the accuracy improves. We also observe that when the number of words is more than 100, the accuracy of using SC and NNSC is higher than that of k-means. While this observation is similar to the case where χ^2 -SVM is used as the classifier, using intersection-SVM is a better alternative when the *DMLSmartActions* dataset is expanded. This is because intersection-SVM in contrast to χ^2-SVM can be efficiently approximated [30] and scales well for large datasets.

For both, χ^2-SVM and Intersection-SVM, using SC and NNSC, results in a better performance than the one using k-means for constructing visual words. The best performance of SC is better than the best achieved accuracy of NNSC.

Figure 5 shows the accuracy of the human action recognition algorithms that use k-means, SC, and NNSC to construct words and a linear –SVM classifier to classify the actions within the chosen dataset. The case where SC is used with linear –SVM classifier is the proposed approach by [8], and the one where NNSC is used with linear–SVM classifier is the suggested algorithm by [9]. As it can be observed from Figure 5, the best accuracy is achieved when NNSC is used for constructing visual words.

Table I reports the accuracy of the tested human action

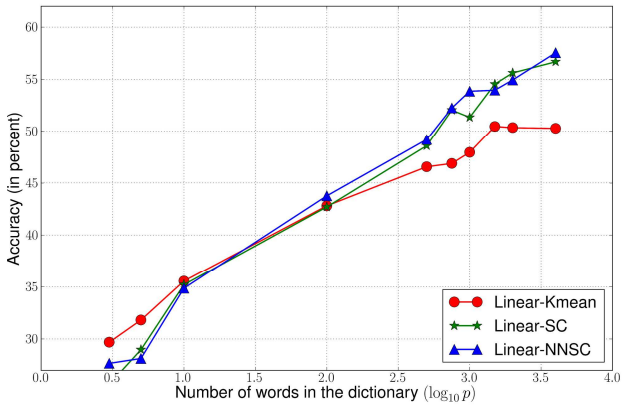

Fig. 5. Recognition accuracy when a Linear-SVM is used as the classifier.

recognition algorithms using the *DMLSmartActions* dataset. As it can be observed, using SC with χ^2-SVM, NNSC with χ^2-SVM, and NNSC with Linear-SVM produce comparable accuracy human action recognition results. In contrast with linear-SVM, χ^2-SVM cannot scale very well for large datasets. Thus, NNSC with Linear-SVM can be used as an alternative for SC with χ^2-SVM, when there are limited computational resources available and the dataset is large (the training for the current size of *DMLSmartActions* dataset is more than 10 times faster when the linear-SVM classifier is used instead of the χ^2-SVM classifier). Note that the accuracy of the proposed algorithms in [7], [1], and [9] is much higher when they are applied on KTH and IXMAS datasets (see [9]). This is because KTH and IXMAS are much simpler than the *DMLSmartActions* dataset. Given the above results, in our future work we plan to develop more effective human action recognition algorithms specifically designed for the smart home environment which will take into account the temporal correlation between different actions (natural order of actions), the scene context, and depth information. Moreover, we plan to expand the *DMLSmartActions* dataset to include a variety of objects, scenes and actions for smart home applications.

V. CONCLUSION

In this paper we introduce a new dataset for human actions in a smart home environment called *DMLSmartActions*. This dataset is specifically collected for helping us analyze human actions in a home environment. Using this dataset, we studied the performance of some existing human action recognition algorithms which had shown excellent performance on other simple datasets. As our experimental results show that the complexity and variations of this dataset make action recognition more challenging that it proved to be when using simple datasets. The low performance of the tested human action recognition algorithms on this dataset suggests revisiting the action recognition problem for smart home applications. Our experiments show that the χ^2-SVM classifier can produce the highest accuracy when visual words are constructed using SC and NNSC. In cases where limited memory and processing power is available, we recommend to use linear-SVM instead of χ^2-SVM, since it can produce comparable results (0.65% lower in accuracy) and requires much less processing power for training.

ACKNOWLEDGMENT

This work was supported in part by NSERC under Grant CRDPJ 434659 - 12 & the ICICS/TELUS People & Planet Friendly Home Initiative at UBC.

REFERENCES

[1] Y. Hao and R. Foster, “Wireless body sensor networks for health monitoring applications,” Physiological measurement, vol. 29, no. 11, p. R27, 2008.

[2] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, “Mobile health monitoring system based on activity recognition using accelerometer,” Simulation Modelling Practice and Theory, vol. 18, pp. 446–455, 2010.

[3] C. Schuldt, I. Laptev, and B. Caputo. “Recognizing human actions: a local SVM approach”. In ICPR, 2004

[4] D. Weinland, R. Ronfard, and E. Boyer, “Free viewpoint action recognition using motion history volumes,” in CVIU, November 2006, pp. 249–257.

[5] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” in IEEE CVPR, 2009.

[6] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quenot, “Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics,” in Proceedings of TRECVID 2012. NIST, USA, 2012.

[7] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in BMVC, 2009.

[8] Y. Zhu, X. Zhao, Y. Fu, and Y. Liu, “Sparse coding on local spatialtemporal volumes for human action recognition,” in ACCV, 2010.

[9] S. M. Amiri, P. Nasiopoulos, and V. C. Leung, “Non-negative sparse coding for human action recognition,” in ICIP, 2012.

[10] A. Bobick and J. Davis, “The recognition of human movement using temporal templates,” IEEE TPAMI, 2011.

[11] T. Syeda-Mahmood, M. Vasilescu, and S. Sethi, “Recognizing action events from multiple viewpoints,” in Event Video, 2001.

[12] I. Laptev and G. Mori, “Statistical and structural recognition of human actions,” in Tutorial in ECCV, 2010.

[13] I. Laptev and T. Lindeberg, “Space-time interest points,” in ICCV, 2003.

[14] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in CVPR, 2009.

[15] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, “Convex optimization with sparsity-inducing norms,” in OPT, 2011.

[16] Y. Boureau, F. Bach, Y. L. Cun, and J. Ponce, “Learning mid-level features for recognition,” in CVPR, 2010.

[17] DmlSmSrtaction: Dataset for human actions in smart environments.[Online].Available: http://ece.ubc.ca/~mohsena/dmlsmartaction.html

[18] I. Laptev, “On space-time interest points,” in IJCV, 2005.

[19] A. B. H. Lee, R. Raina, and A. Y. Ng, “Efficient sparse coding algorithms,” in NIPS, 2007.

[20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” JMLR, 2010.

[21] D. Arthur and S. Vassilvitskii, “k-means++: the advantages of careful seeding,” in SODA 2007.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python ,” JMLR 2011.

[23] S. Maji, A. C. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” in CVPR, 2008.