A SIMILARITY MEASURE FOR ANALYZING HUMAN ACTIVITIES USING HUMAN-OBJECT INTERACTION CONTEXT

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

Understanding the context of human-object interactions plays an important role in human activity recognition. Modeling the interaction context is a challenging problem due to the large number of possible objects in the scene and the large number of ways these objects may seem to relate to human activities taking place in the scene. In addition, providing labeling information of the object and human body parts is a very difficult and labor intense part of the training process. In this paper, we use a new class of kernels for image/video data as an extension of string kernels for 2 and 3 dimensional signals to model the human body parts and objects interaction context. In contrast to similar works, the proposed method does not require labeling of the human body parts and objects in the scene for the learning process, making it more practical when dealing with large datasets. Our experimental results show that the proposed kernel efficiently models the context of human-object interactions in image/video sequences and results in improved performance when compared to state-of-the-art methods.

1. INTRODUCTION

Human action recognition has received a vast amount of attention from the research community in the past decade. Having an accurate action recognition algorithm plays an important role in a wide range of applications, such as video surveillance, occupant monitoring in smart homes, or even in social networks and search engines [1]. The main bottleneck in human action recognition systems is training a model to discriminate between various actions while variations in the action style, surrounding objects and background are present.

In analyzing human actions in a video sequence, motion information is the most commonly visual cue used. In addition to motion cue, other visual cues have been used for action recognition, such as body structure and pose [2-4] as well as interactions between body-parts and surrounding objects [5, 6]. It has been shown that modeling the context of human-object interactions plays an important role in the human visual systems operation [7]. Using human-object interaction cues for modeling human action results in achieving considerable improvements [5, 8-10], especially in the case of human action recognition using still images, where the motion cue is not available [5, 6].

In this work, we propose a learning model, which uses an extension of a string kernel [11] to measure the similarity between two images or two video clips based on interactions between different human body parts and objects. One important contribution of this method is that unlike other previously proposed methods, body-part and object annotations are not directly used in the action modeling procedure, making this approach more affordable to train the algorithm for new large datasets. Performance evaluations showed that our method outperforms existing state-of-the-art methods when applied to elaborate datasets that represent realistic life-like scenarios resulting in true and valuable human action recognition.

The rest of the paper is organized as follows: Section 2 provides an overview of existing methods, Section 3 introduces our proposed approach, Section 4 discusses our experiments for evaluating the performance of our proposed approach, and conclusions are drawn in Section 5.

2. OVERVIEW OF EXISTING METHODS

Recently, human-object interaction cues have received a lot of attention when designing systems for understanding human actions in a still image [5, 8-10]. For example, in a *drinking* scene, it is very probable to observe a *human-hand* or *human-head* close to a *cup* or a *glass*. In this regard, Delaitre *et. al.* use pre-trained body-part and object detectors (instead of using standard local features such as HOG) to build a co-occurrence model and a sparse SVM for classifying human actions in still images [5]. Due to the occlusion and the small size of the objects in a scene, the output of the object and body part detectors is usually noisy and methods that are only based on detectors (e.g., [5]) may not be able to correctly detect all the elements in the image. To solve this problem, Yao *et. al.* propose a method to simultaneously detect objects, human-body parts, human poses, and human actions in the input image [6]. The main idea behind this is that object locations and human poses are correlated and, thus, recognizing one facilitates the recognition of the other and vice versa. Consequently, these priors can result in a better action classifier. A conditional random field is used in [6] to model this correlation and use it human action recognition.

In contrast to still images, human-object interaction cues have not been used widely for human action recognition in video sequences. In [8], Prest *et. al.* propose an approach for modeling human actions using human-object interactions in realistic videos. Their proposed method first extracts features from the video frames and then uses a human detector and object detectors to detect the human and objects in the scene. Using a tracker, the algorithm tracks detected objects and humans. Finally, the relative motions of humans and objects in a video sequence are used as a

	c-a	*c-t*	*a-t*	*b-a*	*b-t*	*c-r*	*a-r*	*b-r*
φ(cat)	λ^2	λ^3	λ^2	0	0	0	0	0
φ(car)	λ^2	0	0	0	0	λ^3	λ^2	0
φ(bat)	0	0	λ^2	λ^2	λ^3	0	0	0
φ(bar)	0	0	0	λ^2	0	0	λ^2	λ^3

Figure 1. The feature space for the string kernel where the dataset has four data points {*cat, car, bat, bar*} and λ=*2*.

highly discriminative cue for learning to recognize the human actions in the scene.

Object detectors and human body parts detectors are the essential parts of many human-object interaction modeling techniques in both image and video signals [5, 6, 8-10]. Using such detectors, though, in order to recognize human actions in video sequences is not straightforward. The first challenge is that applying a large number of detectors on a video sequence is a computationally intensive task, which in turn limits the feasibility of using these algorithms for real-time applications . In addition, training accurate detectors for human body-parts and objects in the scene requires large amount of labeled data, a highly labor-intensive task. Although large labeled-image datasets are available, such as ImageNet [12] and Pascal VOC [13], it has been shown that detectors trained with still images have limited performance in dealing with video data [14]. In this work, we propose an extension for string kernels to model the human-object interaction in image and video signals.

3. PROPOSED METHOD

Many machine learning techniques such as neural networks and decision trees operate on data points after transferring them into a feature space (i.e., feature extraction) and build a hypothesis on the data representation on the feature space. Mapping the data into the feature space for many signals such as text, image and video is not a trivial task and many feature extraction techniques have been proposed to deal with this problem.

Kernel methods provide an alternative approach for mapping the data into the feature space. Kernel methods work based on kernel functions, which calculate the inner product (which can be interpreted as a similarity measure) of two samples of the data in a high dimensional space (feature space). Many learning algorithms (such as Support Vector Machine (SVM)) can be re-written in a way that data points only need to appear inside of the kernel function, hence there is no need to map the data into the feature space implicitly [15].

Bag-of-Words (BoW) is a simple and well known approach for mapping the text data into a high dimensional feature space, which can also be modified to be used with visual data. The main bottleneck of BoW-based approaches is that BoW ignores all the ordering information between words in the sequence or between visual words in visual data. Many techniques such as *Pyramidal Spatial Matching* have been proposed to deal with this issue and preserve the spatial information [16].

In the following subsections, we describe our proposed method in detail. Since the proposed method can be describe as an extension for string kernels, we first describe in subsection 3.1 the string kernel proposed in [11]. Then, in subsection 3.2, we introduce an extension for this string kernel for still images (2-D signals) and video sequences (3-D signals).

3.1. String Kernel

Many signals such as human speech, web data, and DNA sequences can be modeled as strings and text data. In [11], Lodhi

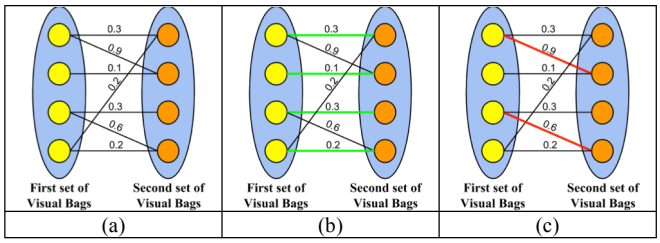

Figure 2. An example of modeling two images as a weighted bipartite graph. (a) Where no matching is applied. The sum over all weights is 2.6; however some vertexes are selected more than one time. (b) demonstrates the maximum cardinality matching. Four nodes are selected on each side, and the sum of the selected weights is 0.9, where two heaviest edges are not selected (c) demonstrates the maximum weight matching of the graph. Two nodes are selected on each side, and sum of the weight is 1.5.

et. al. propose a feature space for analyzing text data, which is the set of all non-contiguous substrings with k-symbols (for k=1, the model is reduced to the Bag of Word model). Moreover, they propose an efficient algorithm for calculating the kernel (inner product) for this feature space without direct mapping of the text data into the feature space. In this proposed feature space, two text sequences are more similar when they have more substrings in common.

This method considers non-contiguous substrings and to deal with non-contiguous substrings a decay factor, $\lambda \in (0, 1)$, is added to the model. In this model, λ^n is used as the weight of a certain feature (substring) in the text, where n is the length of the substring. This means that having more non-common symbols in two substrings reduces their similarity. For example, assume that we have a text dataset which contains four strings (*cat*, *car*, *bat* and *bar*). For k=*2* (all substrings with length of two), we have eight substrings in the dataset as shown in Figure 1. Figure 1 shows the representation of each word in this feature space, where *φ(X)* is the representation for the input in the feature space and λ is a decay factor between zero and one. The coefficient of substring *c-a* in *φ(cat)* is λ^2, which means the length of *c-a* in *cat* is two and the coefficient of substring *c-t* in *φ(cat)* is λ^3, which means the length of *c-t* in *cat* is three. In total, *φ(X)* in this space has a length of $\sqrt[2]{\lambda^4 + \lambda^6 + \lambda^4}$ (*c-a* is shorter and has a larger weight than *c-t*).

The kernel function *K(X,Y)* for this feature space is the inner-product of the mappings *φ(.)* of the inputs:

$$K(X,Y) = \langle \phi(X), \phi(Y) \rangle \quad (1)$$

As suggested in [11], the kernel function can be normalized, $\hat{K}(X,Y)$, to produce higher prediction accuracies as follows:

$$\hat{K}(X,Y) = \langle \hat{\phi}(X), \hat{\phi}(Y) \rangle = \langle \frac{\phi(X)}{\|\phi(X)\|}, \frac{\phi(Y)}{\|\phi(Y)\|} \rangle \quad (2)$$

In next subsection, we introduce our proposed kernel function as an extension of this string kernel for 2-D (image) and 3-D (video) signals.

3.2. Extended String Kernel

In our proposed method, a kernel function is designed for measuring the similarity between two images or two video clips. This kernel function captures local co-occurrence of the patches in the data point, which can be interpreted as human-object interaction in human action recognition applications.

To calculate the proposed kernel function, in a similar fashion to BoW, we first detect feature points and we extract feature vectors (i.e., SIFT or SURF features can be used for image data,

and HoG3D or STIP features can be used for video data). Then, using an unsupervised clustering algorithm (e.g., *k-means*), we can build a codebook and construct visual words.

In the second step, a sliding window (width of *W*) is moved through the image or video clip. At each position, the collection of visual words is collected as mini-visual-bags (MVB). Note that we consider these mini-visual-bags as sets of visual words, so we only keep one version of each set and ignore the relative geometry of the visual words in the set. This gives some degree of rotation/scale invariance to the similarity measure.

By collecting all MVBs for each data point, we can model the information regarding to the context of the spatial interaction between different parts of an image or the context for spatiotemporal interactions between different parts of a video clip. We can define a similarity measure between two data points as follows:

$$K_s(X_i, X_j) = \sum_{\forall M_l \in X_i} \sum_{\forall M_k \in X_j} SIM(M_l, M_k) \qquad (3)$$

where X_i and X_j are two sets of MVBs and $SIM(M_l, M_k)$ is a similarity measure between two MVBs. Since MVBs are sets of visual words, a similarity measure between two sets can be used to calculate *SIM*. In particular, the similarity of two sets in this context is the relative size of the intersection of the sets, which is similar to the decay factor in the string kernel in [11]. In this paper, we use *Jaccard index* (*Jaccard Similarity coefficient*) for measuring the similarity of two sets, which can be calculated as the ratio of the cardinality of sets intersection and sets union:

$$SIM(M_l, M_k) = \frac{|M_l \cap M_k|}{|M_l \cup M_k|} \qquad (4)$$

Since the number of MVBs changes between different data points, the value of $K_s(X_i,X_j)$ varies in a large range. Therefore, as suggested in [11], we use a normalized version of the similarity measure $\acute{K}_s(X_i, X_j)$ for generating the results:

$$\acute{K}_s(X_i, X_j) = \frac{K_s(X_i, X_j)}{\sqrt[2]{K_s(X_i, X_i).K_s(X_j, X_j)}} \qquad (5)$$

For this type of kernels, it is easy to show that Mercer's theorem is satisfied, because the kernels are defined directly based on an inner product.

Our proposed similarity measure between two sets of MVBs in (3) calculates the sum over all pairwise similarity measures of MVBs belonging to X_i and X_j. This means that in (3) the similarity of each MVB in X_i appears multiple times in $K_s(X_i, X_j)$. However, it is more reasonable to assume that each MVB appears at most one time in the calculation of the kernel, because one MVB can at most be matched with one MVB in the other image/video. To satisfy this notion, we propose a new similarity measure based on the *Maximum Weight Matching* between two sets of MVBs. In this approach, we build a bipartite graph, in which vertexes in the left side of the graph represent the MVBs in the first set and vertexes in the right side of the graph represent MVBs in the second set. In this graph, the weight of the edge between two vertexes M_l (in the left side) and M_k (in the right side) is defined as $SIM(M_l, M_k)$. By finding the maximum weight matching for this graph, each MVB in the first data point will be matched to at most one MVB in the other data point. Figure 2 demonstrates different ways of calculating the proposed similarity measure. In this figure, each vertex in the left side represents one MVB in the first image and each vertex on the right side represents one MVB in the second image. Edges in this graph represent the similarity scores (SIM) between different MVBs from the first image and the second image. Please note that edges with weight zero are not shown in this graph.

The maximum weight matching problem can be written as a mixed integer linear program:

In general, solving a mixed integer linear program is a NP-hard problem, while the best known exact algorithm for solving the maximum weight matching is $O(n^3)$, where n is the number node (visual mini-bags) in the graph [17].

In the scale of our problem (where n is typically in the range of 500 to 2000), it is not feasible to use an $O(n^3)$ algorithm. Thus,

$$K_m(X_i, X_j) = Max \sum_{\forall M_l \in X_i} \sum_{\forall M_k \in X_j} m_{lk}.SIM(M_l, M_k) \qquad (6)$$

Subject to:

$$m_{lk} \in \{0, 1\}$$
$$\forall M_l \in X_i \sum_{\forall M_k \in X_j} m_{lk} \leq 1$$
$$\forall M_k \in X_j \sum_{\forall M_l \in X_i} m_{lk} \leq 1$$

to report the results in this paper, we use a 2-approximation greedy algorithm [18] to calculate the new similarity measure, $K_m(X_i, X_j)$, based on the *maximum weight matching* of the bipartite graph.

4. EXPERIMENTAL RESULTS

In our study, we investigate the performance of our proposed method and compare it with the state-of-the-art human action recognition methods. In subsection 4.1 we provide more details on the implementation aspects of our experiments while in subsection 4.2 we present the databases we used for our experiments and compare the results of our approach with those of other methods.

4.1. Implementation details

In this work, we use the SURF (Speeded Up Robust Features) [19] feature as implemented in the OpenCV library for extracting features from still images. STIP [20] from the executable provided by [21] is used for feature extraction in video clips. The SPAMS (SPArse Modeling Software) library is used for NNSC implementations [22]. The number of visual words in the NNSC dictionary is fixed to 2000 (for the Sport Dataset [23]) and 8000 (for the other two datasets). For other parameters we follow the recommendations in [24]. An implementation of the Support Vector Machine (SVM) and *k-means* from the Scikit-learn library [25] are used. For learning the codebook for the proposed kernel, we search for the size of the codebook in the range of 100 to 1000 and report the highest accuracy in each experiment.

4.2. Datasets and experimental results

In this paper, we evaluate our proposed method using three human action datasets, one dataset of still images and two datasets of video sequences.

5.2.1. The DMLSmarthome dataset [26]: The DMLSmartActions dataset is the most comprehensive and challenging dataset for human action recognition in a home environment. This dataset contains twelve different actions performed by 17 subjects (6 females and 11 males), which are common in people's day-to-day life in home environments (*clean-table, drink, drop-and-pickup, fell-down, pick-something, put something, read, sit-down, stand-up, use-cellphone, walk, and write*). This dataset contains complex actions, a large number of subjects, and complex backgrounds compared to the Gupta Video dataset (also tested - see below) [28]. For this experiment, we use only the high-definition streams in the dataset (leaving out the depth and RGB information). In our tests we use a combination of the method presented in [24] and a SVM

with the proposed kernel function. For fair comparison with other methods presented in [24, 26], we use LOO (Leave-One-Out) for cross validation. The window size for calculating the kernel is 32x32 in the spatial domain and 15 frames in the temporal domain. Table 1 shows the accuracies achieved by the different algorithms using this dataset. We observe that the highest accuracy is achieved when the proposed human object interaction model is combined with the approach from [24]. Note that our method outperforms even the Meta Learning method by 2.7%, despite the fact that the latter is also using depth and RGB information (in addition to HD data). This is a significant achievement as the dataset used here is very detailed, representing very accurately a significant number of life-like actions found in a smart home environment. We believe that by using the depth and RGB information in the future, our method has the potential to achieve much higher accuracy.

4.2.2. Gupta Video dataset [28]: This video dataset contains six actions (*drinking-from–a-cup, spraying-from-a-bottle, answering-a-phone-call, pouring-from-a-cup, and lighting-a-flashlight*). Once more, to report the results, we use a combination of the method from [24] and a SVM with the proposed kernel function. The window size for calculating the kernel for this dataset is 16x16 in the spatial domain and 8 frames in the temporal domain. Table 2 demonstrates the achieved accuracies by two of the state-of-the-art algorithms (presented in [8] and [28]) on this dataset. We observe that in this case, the overall accuracy of our proposed method matches that of two other state-of-the-art methods [8, 28]. The reason for having equal (and very high) accuracy is that this dataset is a fairly simple dataset compared to the DMLSmartActions dataset [26], and all methods manage to accurately recognize a set of simple actions. This supports our claim that the proposed method can capture the useful information in the video signal without accessing extra labeling information and at the same time despite its simplicity not sacrificing accuracy.

4.2.3. The Sport dataset [23]: The sport dataset is used for evaluating the proposed kernel function for modeling human-object interaction in still images. This dataset contains six sport activities (*tennis-forehand, tennis-serve, volleyball-smash, cricket-defensive shot, cricket-bowling and croquet-shot*) and fifty examples from each action. We chose to test our method for still images to show its versatility and capability to be a promising foundation for a much more advanced and accurate method in the future. For a fair comparison with the performance of the proposed algorithms in [6, 23], we follow the recipe from [23] and use a 30-20 split for training and testing. Our proposed method outperforms all the other methods by a range of 1.6% to 34.6%. It is worth noting that the algorithms presented in [6, 23] have access to the object and human body part annotations, while our proposed method did not use this annotation data. Table 3 shows the achieved accuracies for the sport dataset [23]. In this case, we use the combination Spatial Pyramid Matching [16] with a grid size of 3x3 and non-negative sparse coding and max-pooling, plus a SVM with a the proposed kernel function with maximum weight matching and window size of 16x16. We observe that our technique outperforms the state-of-the-art methods for still images for this dataset. The state-of-the-art method [31] uses a latent SVM [29] for learning the actions, which explicitly models the human object interactions. Overall, our method shows to be a promising new approach and has the potential to lead to higher accuracy if additional data such as depth and skeleton information is used in a future more advanced implementation.

TABLE 1. Achieved accuracies of different algorithms on the *DMLSmarthome dataset* [26]

Algorithm name	Accuracy
SVM with k-means [30]	54.7%
Meta Learning (RGB+Depth) [27]	77.2%
SVM with NNSC[24]	58.2%
SVM with the proposed kernel	62.6%
Overall (SVM-NNSC + Proposed Kernel)	**79.9%**

TABLE 2. Achieved accuracies of different algorithms on the *Gupta Video dataset* [23]

Algorithm name	Accuracy
Gupta *et. al.*[28]	93%
Prest *et. al.* [8]	93%
SVM with NNSC[24]	83%
SVM with the proposed kernel	87%
Overall(SVM-NNSC + Proposed Kernel)	**93%**

5. CONCLUSION

In this work, we proposed a new human action recognition approach that is based on using a kernel function for measuring the similarity of two images or two video sequences by calculating the similarity of the interactions between objects and human body parts. The main bottleneck for applying previously proposed methods for modeling these interactions is that they usually need to have access to labeling information for object and body parts in the input space. The main contribution of the proposed method is that is independent of this labeling information. Thus, in situations like action recognition for video signals where providing such information is very difficult and labor intensive, the proposed method still can model actions based on interactions between human body parts and their surrounding objects.

We examine our proposed method on one image dataset and two video datasets. Our experimental results drawn from using these datasets, support our method outperforms the existing state-of-the-art methods when applied to elaborate datasets that represent realistic life-like scenarios resulting in true and valuable human action recognition.

6. ACKNOWLEDGEMENT

This work was made possible by NPRP grant # NPRP 4-463-2-172 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

TABLE 3. Achieved accuracies of different algorithms on the Sport dataset [23].

Algorithm name	Accuracy
Yao *et. al.* [6]	83.3%
Gupta *et. al.* [23]	78.9%
Delaitre *et. al*. [31]	85.1%
Bag of Words (Linear SVM)	52.5%
Spatial Pyramid Matching (SPM)[16]	73.3%
SVM with the proposed kernel	67.5%
Overall (SPM+Proposed Kernel)	**86.7%**

6. REFERENCES

[1] B. Ni, Y. Song and M. Zhao, "YouTubeEvent: On Large-Scale Video Event Classification," IEEE ICCV'2011.

[2] B. Sapp and B. Taskar, "MODEC: Multimodal Decomposible Models for Human Pose Estimation," In Proc. CVPR 2013.

[3] B. Rothrock, S. Park, S. C. Zhu, "Integrating Grammar and Segmentation for Human Pose Estimation," In Proc. CVPR 2013.

[4] S. Maji, L. Bourdev and J. Malik, "Action Recognition from a Distributed Representation of Pose and Appearance," In Proc. CVPR 2011.

[5] V. Delaitre, J. Sivic and I. Laptev, "Learning person-object interactions for action recognition in still images," NIPS, 2011.

[6] B Yao and L. Fei-Fei, "Recognizing Human-Object Interaction in Still Images by Modeling the Mutual Context of Objects and Human Poses," IEEE TPAMI, Vol. 34, No. 9, 2012.

[7] I. Biederman, R. Mezzanotte and Rabinowitz, "Scene Perception: Detection and Judging Objects Undergoing Relational Violations," Cognitive Psychol., Vol 14, 1982.

[8] A. Prest, V. Ferrari and C. Schmid, "Explicit Modeling of Human-Object Interaction in Realistic Videos," IEEE TPAMI, Vol. 35 No. 4, 2013.

[9] B. Yao and L. Fei-Fei, "Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions," In Proc. CVPR 2010.

[10] A. Prest, C. Schmid and V. Ferrari, "Weakly Supervised Learning of Interactions between Human and Objects," In IEEE TPAMI, Vol. 34, No. 3, March 2013.

[11] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini and C. Watkins, "Text classification using string kernels," Journal of Machine Learning Research, 419–444, 2002.

[12] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," In Proc. CVPR 2009.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[14] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller, "Shifting Weights: Adapting Object Detectors from Image to Video," NIPS, 2012.

[15] B. Schölkopf and A. J. Smola, "A Tutorial Introduction," in "Learning with Kernels," The MIT Press, 2002.

[16] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," In Proc. CVPR 2006.

[17] Z. Galil, "Efficient Algorithms for Finding Maximum Matching in Graphs," In ACM Computing Surveys (CSUR), Vol. 18, No. 1, 1986.

[18] R. Preis, "2-Approximation Algorithm for Maximum Weighted Matching in General Graphs," Symposium on Theoretical Aspects of Computer Science (STACS) 1999.

[19] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "SURF: Speeded Up Robust Features," CVIU, Vol. 110, No. 3, 2008.

[20] I. Laptev and T. Lindeberg, "Space-Time Interest Points," in Proc. ICCV, 2003.

[21] I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld, "Learning Realistic Human Actions from Movies," in Proc. CVPR 2008.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," JMLR, 2010.

[23] A. Gupta, A. Kembhavi and L. Davis, "Observing Human-Object Interactions: Using Spatial and Functional Compatibility for Recognition." IEEE TPAMI, Vol 31, No.10, October 2009.

[24] S. M. Amiri, P. Nasiopoulos, and V. C. Leung, "Non-Negative Sparse Coding for Human Action Recognition," In Proc. ICIP, 2012.

[25] 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.

[26] S. M. Amiri, M. T. Pourazad, P. Nasiopoulos and V. C.M. Leung, "Non-Intrusive Human Activity Monitoring in a Smart Home Environment," in Proc. IEEE HealthCom 2013

[27] S. M. Amiri, M. T. Pourazad, P. Nasiopoulos and V. C.M. Leung, "Human Action Recognition using Meta Learning for RGB and Depth Information," In Proc. IEEE ICNC 2014.

[28] A. Gupta and L. S. Davis, "Objects in Action: An approach for Combining Action Understanding and Object Perception," In Proc. CVPR 2007.

[29] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," In IEEE TPAMI, Vol. 32, No.9, September 2010.

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

[31] V. Delaitre, I. Laptev and J. Sivic "Recognizing human actions in still images: a study of bag-of-features and part-based representations," In Proc. BMVC 2010.