# Activity Recognition from Wearable Cameras

Panagiotis Giannakeris
*ITI-CERTH*
Thessaloniki, Greece
giannakeris@iti.gr

Konstantinos Avgerinakis
*ITI-CERTH*
Thessaloniki, Greece
koafgeri@iti.gr

Stefanos Vrochidis
*ITI-CERTH*
Thessaloniki, Greece
stefanos@iti.gr

Ioannis Kompatsiaris
*ITI-CERTH*
Thessaloniki, Greece
ikom@iti.gr

***Abstract*—A novel work for Ambient Assisted Living applications is presented here. More specifically, this paper focuses on activity recognition from recordings of daily living captured by wearable cameras. It constructs a discriminant object centric motion descriptor for representing the micro-actions within the viewpoint of the action maker so as to later define the activity that he/she performs. The accumulation of these activities build patterns over time that can be used to study the behavior of the end-users, which is very useful for health application and monitoring of patients inside their own dwellings or their behavior inside a controlled environment.**

***Index Terms*—Activity recognition, Object detection, Egocentric vision, Ambient assisted living.**

## I. Introduction

Nowadays, more and more patients that do not have a critical disease are called to live inside their own homes, as nursing homes and hospitals can not accommodate them in their own premises for too long. However, for some of them it is essential that a doctor or a carer should continue monitor their health and keep a log file of their behaviors throughout time. This work is motivated by this need and proposes an unobtrusive and efficient way to gather visual data of human patients activities of daily living, so that it could build their behavioral pattern throughout time and alleviate the workload of their carers and doctors. Except from that, our system keep a log file of the objects that a human patient uses and their position inside a house, so that they could be informed about anytime they want.

Activity recognition of daily living is a very hot topic amongst the computer vision domain and a lot of works have been proposed in the last decade to solve this challenge. Many of them propose to describe activities by an object centric manner following the information that derives from the existence of specific objects in the scene [6], [15], [13], [21]. Moreover, scene understanding is also used in [17]. Other works leverage the motion that appears in the scene and extract features so as to represent the activities that take place [11], [12]. In [20] a multi-task clustering framework tailored to first-person view (FPV) activity recognition is presented. Another more recent approach is to use deep CNN architectures [19] to learn deep appearance and motion clues. Deep CNNs are also used to learn hand segmentations in order to understand the activities that a user performs and his interaction with other users that might also appear in the video frame [21], [2], [1]. More recent works focus on multi-modal analysis of egocentric cameras and information from other wearable sensor equipment with the deployment of early or late fusion schemes [14], [4], [3].

In contrast with the aforementioned methods we not only detect relevant objects but also extract their individual motion patterns and group them by class over short temporal windows. Subsequently, we encode those patterns in a binning framework to understand their usage in short term actions which are fundamental building blocks of long term activities. In addition, unlike other State-of-the-Art (SoA) works, our current implementation does not rely on any hand movement information at all or other sensor equipment. Results in a benchmark dataset shows the capability of the proposed work by profoundly evaluating in a great deal of parameters and comparing it with SoA work.

The rest of this paper is organized as follows. In section II, methodology is presented, while in section III the experimental results are included. Finally, conclusions are drawn in Section IV.

## II. Methodology

In order to successfully recognize activities of daily living such as ”book reading”, ”hand washing” or ”preparing breakfast” that take place inside an egocentric video, it is important at first to get a deep understanding of the short time lower level actions a person is performing sequentially in order to accomplish the bigger scale ones. For example the activity ”preparing breakfast” involves the short time actions ”opening the fridge”, ”grabbing butter”, ”closing the fridge”, ”taking a knife”, ”spreading the butter” etc. This group of micro-actions as we call them, does not always need to form a complicated sequence for every activity. For example the activity ”reading a book” besides the actual reading usually involves only one micro-action performed repeatedly: ”turning the page”. For those reasons we seek a way of extracting a representation of the full duration of an activity video that will be informative towards the set of micro-actions that are included and have a strong ability to uniquely describe the activity.

This work was supported by V4Design and SUITCEYES projects partially funded by the European Commission under grant agreements No 779962 and No 780814, respectively.

978-1-5386-7021-7/18/$31.00 

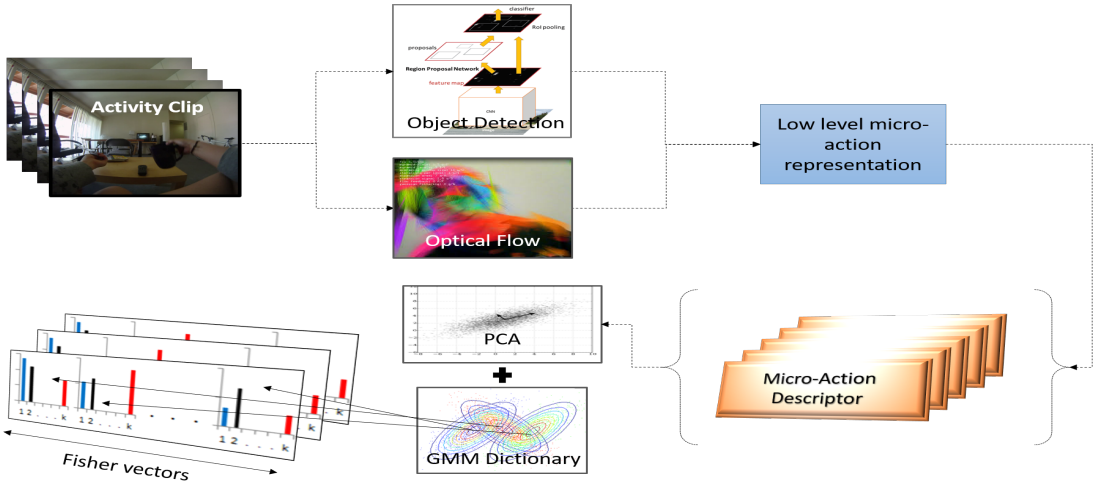


Fig. 1. Block diagram of the proposed methodology.

It is also very well established [6] [15] [13] [21] that every activity is related closely to a group of active objects and a group of passive objects. The first group contains objects that are handled by the person during the activity and the second group contains objects that are simply within view of the camera when the activity is performed. Objects are good indicators of certain activities such as the TV in the "watching television" activity or the book in the "reading a book" activity. We further elaborate this notion by hypothesizing that not only the presence but also the characteristic motion of the objects that is taking place in the scene is powerful enough to discriminate between active and passive ones and at the same time inform about the activities that are happening. For example motion information from dishes that are being washed combined with the presence of a tap in the scene can uniquely describe the "washing dishes" activity.

The above assumptions are taken into account in our activity recognition method. The overall framework is shown in figure 1. First we detect objects using a deep CNN architecture that combines deep feature extraction network and a bounding box coordinate regression network that predicts object classes and locations in the video frames. We combine the powerful detector with a tracking algorithm eliminating the need to utilize the deep architecture for every frame in order to achieve near real time object detection. Then, every detected object's motion is analyzed using HOF or MBH [5] features so as to form the lower level micro-action representations that appear in short time windows over the full activity sequence. Finally GMM clustering of the micro-action descriptors is performed in order to find the most discriminative of the full set. Given a set of micro-action descriptors extracted for a single activity sequence and the GMM clustering centers, a Fisher encoding scheme is used in order to yield the final descriptor of the full activity sequence in a Bag-of-Micro-Actions type of representation.

### A. *Object Detection*

For the purpose of detecting objects of interest in video frames we chose to extract deep image representations from a CNN and predict pixel coordinates of bounding boxes using a deep CNN object detector. To this end we adopt a modification of the accurate Faster-RCNN that was originally proposed in [16]. A thorough evaluation of this model and comparisons with other SoA deep object detectors presented in [9] reveal that the Faster-RCNN-resnet101 architecture achieves a good trade off between speed/accuracy. This model incorporates the resnet101 [7] deep feature extractor and a region proposal network along with a bounding box classifier and coordinate regressors. We chose this architecture because it achieves very fast object detection by using a single feed-forward convolutional network to directly predict classes and bounding boxes of objects. In order to speed up our object detection procedure during inference time we find useful to track the detected objects found in a frame into the next $T$ frames of the video. By assigning a detection rate of $T > 15$ our combined detector and tracker algorithm achieves real time performance. We manually set the detection rate parameter to 15 following empirical evaluation after trials with other values ranging from 15 to 30. Intuitively, the detection rate defines the temporal resolution of the continuous object detection function. Lower detection rate means higher temporal resolution of the detector and vise versa. Note that by setting the detection rate to 15 the detector only runs once between half-second intervals and the tracker works the rest of the time which yields an adequate temporal resolution considering that it is very unlikely that an object will appear and disappear in less than that time.

The core functionality of our tracker is based on the KCF tracking algorithm that was proposed in [8]. The detector is used initially in order to detect objects every $T$ video frames and initialize the new object candidate database with new entries. Bounding box coordinates are stored over time so that full trajectories can be build. For every new ID its

corresponding class label and a detection score is saved as well. Afterwards, the algorithm checks the new detections from the candidate pool for overlaps with already existing recent trajectories. Then, based on an IoU score check it rejects found boxes that exceed an overlap threshold to avoid creating multiple identities for the same object. Next, we feed the KCF tracker with the remaining boxes in order to localize their position throughout sequential video frames. Future detections of already tracked objects are also utilized in order to rectify the bounding boxes of the monitored objects. When a detection is missed, we relocalize the bounding box relying only on KCF update coordinates, while when the algorithm does not localize any tracked object for *l* sequential video frames the object is presumed to have traveled off the frame. To tackle overlaps between True Positive (TP) cases we chose to merge the trajectories at the current frame and assign the oldest ID to the resulting trajectory.

### *B. Micro-action representation*

In Figure 2 we can see how our proposed object motion descriptors are computed. Our method builds representations of short term low level actions of fixed temporal window $W$ from the motion patterns of the objects that are found in this window. More specifically, we compute dense optical flow to extract the full scene's motion between two consecutive frames. We use the OpenCV implementation of the Fast optical flow using the dense inverse search algorithm proposed in [10]. In addition, doing the calculation every other frame inside the window instead of every frame leads to $W/2$ calculations which yields faster computation times. Having already detected the objects in a particular frame we take each bounding box as our region of interest and crop the dense optical flow map accordingly taking only the portion that belongs to the object. Consequently we can calculate HOF (histograms of optical flow) descriptors that represent an object's motion.

To calculate an object's HOF descriptor we apply a $2X2$ uniform grid on top of the bounding box region. For each one of the 4 cells flow orientations are quantized into an 8 bin histogram weighted by their magnitude values. In addition we chose to apply a soft binning method that distributes the votes between adjacent bins based on the distances of the values from adjacent bins centers. This procedure results in a 32-Dimensional motion descriptor that is extracted for every object class in the scene. If multiple objects from the same class appear in a frame we chose to aggregate the vectors and divide by the number of objects so as to get the average motion descriptor of that particular class. In the case of absence of objects from a particular class the corresponding HOF descriptor is set to the zero vector. Let $C$ be the number of classes the detector can predict, $N_c$ the number of objects found of class $c$. The early object class motion descriptors are formed as follows:

$$D_c = \frac{1}{N_c} \sum_{j=1}^{N_c} HOF_{32}, \quad c = 1 \dots C \tag{1}$$

By concatenating L2 normalized motion descriptors for each class we get a complete description for a pair of consecutive frames in the window $W$:

$$R_f = \{D_1, D_2, \dots, D_c\} \tag{2}$$

Finally, we concatenate those descriptors throughout $W/2$ frame pairs to get a complete representation of a micro-action composed by the object's movement patterns that appeared in the window:

$$M = \{R_1, R_2, \dots, R_{W/2}\} \tag{3}$$

One problem with the accurate extraction of object motion from egocentric videos is that very frequently the wearable camera moves along with the person that is wearing it. As a result global camera motion may overpower the delicate dynamics of the objects' motion that we are trying to capture. Therefore we consider an alternative to the HOF descriptor that is the MBH (motion boundary histogram) descriptor where the optical flow field is first separated into its x and y component and spatial derivatives are computed for each one of them. This time we obtain a 32-dimensional for each component (64-dimensional overall) and we follow the same procedure to obtain the final descriptor as in the HOF descriptor case. Because MBH is the gradient of the optical flow, any motion that is happening constantly (global motion) is suppressed and only information about changes in the flow field (i.e., motion boundaries) is kept [18]. Compared to video stabilization and motion compensation this is a faster method of discarding global motion information.

### *C. Activity recognition*

For a given activity sequence the extraction of micro-action descriptors that represents a small sequence of $W$ frames takes place with a stride of $S$ frames. We chose that value to be exactly 1 second in all our experiments. This simply means that for every micro-action descriptor $M$ we skip 1 second into the video before we begin extracting the next micro-action descriptor. Contrary to using overlapping windows the stride parameter was inserted to give our method a speed boost. Given that the micro-action window $W$ is chosen sufficiently small it is guaranteed that the number of micro-actions for an activity sequence will be enough for the activity to be adequately represented.

In this section we describe how a micro-action vocabulary is trained using the descriptors that have been extracted following the previous section. Subsequently, all micro-action descriptors extracted from all the training activity sequences are fed into a Fisher encoding scheme. This way, a micro-action vocabulary based on the most discriminating ones is constructed. The computation of the most discriminating samples is performed by applying unsupervised clustering (Gaussian Mixture Model (GMM)) in the micro-action representation hyperspace.

Let $\{\mu_j, \Sigma_j, \pi_j; j \in R^L\}$ be the set of parameters for $L$ Gaussian models, with $\mu_j$, $\Sigma_j$ and $\pi_j$ standing respectively for the mean, the covariance and the prior probability weights of the $j^{th}$ Gaussian. Assuming that the $D$-dimensional early

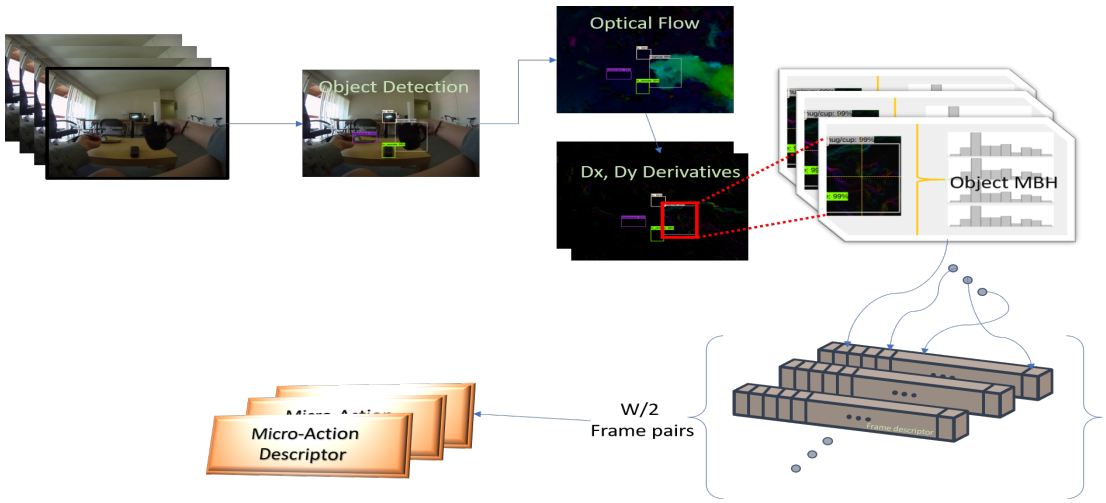


Fig. 2. Computation of Micro-Action descriptor.

descriptor is represented as $\overline{M}_i \in R^D; i = \{1, \ldots, N\}$, with $N$ denoting the total number of descriptors, Fisher encoding is then built upon the first and second order statistics:

$$f_{1j} = \frac{1}{N\sqrt{\pi_j}} \sum_{i=1}^{N} q_{ij} \sigma_j^{-1} (\overline{x}_i - \overline{\mu}_j)$$
$$f_{2j} = \frac{1}{N\sqrt{2\pi_j}} \sum_{i=1}^{N} q_{ij} [\frac{(\overline{x}_i - \overline{\mu}_j)^2}{\sigma_j^2} - 1] \quad (4)$$

where $q_{ij}$ is the Gaussian soft assignment of descriptor $M_i$ to the $j^{th}$ Gaussian and is given by:

$$q_{ij} = \frac{exp[-\frac{1}{2}(M_i - \mu_j)^T \Sigma_j^{-1} (M_i - \mu_j)]}{\sum_{t=1}^{L} exp[-\frac{1}{2}(M_i - \mu_t)^T \Sigma_j^{-1} (M_i - \mu_t)]} \quad (5)$$

Distances as calculated by Eq. 4 are next concatenated to form the final $2LD$-dimensional Fisher vector, $F_X = [f_{11}, f_{21}, \ldots, f_{1L}, f_{2L}]$, that characterizes each activity sequence. The final Fisher encoding for a specific activity sequence can now be classified using an SVM or a Neural Network classifier.

## III. Experimental Results

### A. Dataset

We performed our experiments in the ADL dataset [15]. It is composed of videos recorded with a wearable camera from 20 different persons. The videos contains very realistic scenes of daily living and is challenging due to the existence of global camera motion as a result of the camera movement. The objects are also in many cases occluded. Activity start time and end time annotations and object annotations are available for each one of the 20 videos. From the 48 different classes for objects that are available in the current version of the dataset we select the 34 most frequently annotated to train our object detector. Those are: "tv remote", "tea bag", "towel", "door", "pan", "knife/spoon/fork", "cell phone", "soap liquid", "vacuum", "detergent", "tv", "pills", "tap", "fridge", "blanket", "microwave", "container", "cell", "dent floss", "mug/cup", "person", "toothbrush", "food/snack", "book", "tooth paste", "dish", "trash can", "kettle", "bottle", "comb", "laptop", "pitcher", "oven/stove", "washer/dryer". We also select a subset of 18 activity classes as in [21] to train our activity recognition algorithm so as to follow similar approaches with previous works and present comparable results.

### B. Experimental work

In this section we describe the experiments that we made so as to select the best parameters for our activity recognition algorithm and compare the two proposed descriptors (HOF, MBH) to see their applicability. Furthermore, we accumulated and present activity recognition results for each class for our two best models in the form of confusion matrices. Finally, we present a comparison of our results with other SoA works for the ADL dataset so that we can prove the applicability of our algorithm.

To train our object detector we used the first 6 videos of the ADL dataset setting the detection rate to 15 frames as mentioned previously. We experimented with two different durations for the temporal window: $W = 90$ and $W = 60$ frames. Those two values correspond to 3 seconds and 2 seconds respectively for the videos of the ADL Dataset which were recorded at 30fps. Considering that the activities average duration is in the order of minutes in this dataset, we manage to get enough micro-action descriptors assigned to each activity and simultaneously capture more complex object motions through time. Furthermore, we show that micro-actions of 3 or 2 seconds are long enough for our method to perform close to SoA levels. The two choices for our temporal window $W$ proves to be convenient for algorithmic speed considerations as well. The length of the micro-action descriptor is $\frac{W}{2} \times 34 \times 32$ for the HOF descriptor and $\frac{W}{2} \times 34 \times 64$ for the MBH descriptor.

As a means of dimensionality reduction we perform Principal Component Analysis on our low level descriptors. PCA guarantees maximum variance of the samples in the lower dimensionality space. We chose two possible reductions in our experiments: 80 and 256 components. This way, our early micro-action descriptor's dimensionality reduces from some thousand components to only a couple of hundreds. This also aids the process further down the line as the Fisher encoding scheme will multiply that amount by double the vocabulary size. Additionally, we experiment with two different vocabulary sizes using 32 or 64 words.

For the final step, we deploy as our classifier a fully connected neural network (NN1) with a depth of two layers of width 512 and 256 accordingly, using RELU activations, 50% chance of dropout between layers and softmax activation in the output layer. Another similar architecture (NN2) was also deployed with half the amount of neurons for each layer (256 in the first layer and 128 in the second) and a linear SVM classifier as well.

To evaluate the action recognition performance as in [21], we performed the leave-one-person-out cross-validation method for every parameter combination we discussed and finally we report the per-class average precision (mAP) score. Tables I and II present analytically our scores for every experiment. As it is shown choosing 256 components in PCA results in significant performance boost when combined with a larger temporal window. Choosing 80 components resulted in better performance in some cases of the shorter temporal window. This behavior is somewhat expected since the more lengthy the temporal window becomes, the dimensionality of the micro-action descriptor gets higher and as a result less components must be discarded. Moreover increasing the size of the vocabulary from 32 to 64 failed to improve our results and especially when using the shorter temporal window. This proves that using a smaller vocabulary consisting of 32 words is enough to get good coverage of the most discriminant micro-actions of the entire dataset. Overall, the best models came from the combination of 256 PCA components coupled with a GMM vocabulary of size 32 and the neural network architecture with the most learnable parameters (NN1). Finally, we can see that the MBH descriptor almost entirely outperformed the HOF descriptor for every experiment with a temporal window of 60 frames and that the performance of the two was comparable for a window of 90 frames. This is an indication the MBH has to offer more when micro-action extraction is more refined in time.

In Table III, we compared the accuracy rates of our best models to the ones that are mentioned in the literature. As already described we followed the evaluation procedure in [21] in order to present comparable results. As we can see, the MBH version of our method outperformed every other. The HOF descriptor is also highly ranked.

Next, we select our top two models (one for each descriptor) and train them for the first 6 videos of the dataset. We present the test set confusion matrices in Figures 3 and 4. As we can see, MBH performed better than HOF in most of the classes

TABLE I
ACTIVITY RECOGNITION RESULTS FOR **HOF** DESCRIPTOR

| Model comparison (mAP%) for HOF descriptor | | | |
|---|---|---|---|
| | SVM | NN1 | NN2 |
| W 90 + PCA 80 + GMM 32 | 43.19% | 52.40% | 47.07% |
| W 90 + PCA 80 + GMM 64 | 46.22% | 51.04% | 51.56% |
| W 90 + PCA 256 + GMM 32 | 45.21% | **52.86%** | 51.86% |
| W 90 + PCA 256 + GMM 64 | 46.22% | 51.03% | 51.56% |
| W 60 + PCA 80 + GMM 32 | 43.51% | 50.98% | 48.66% |
| W 60 + PCA 80 + GMM 64 | 43.24% | 47.34% | 44.69% |
| W 60 + PCA 256 + GMM 32 | 46.30% | 48.07% | 47.66% |
| W 60 + PCA 256 + GMM 64 | 45.73% | 46.81% | 45.53% |

TABLE II
ACTIVITY RECOGNITION RESULTS FOR **MBH** DESCRIPTOR

| Model comparison (mAP%) for MBH descriptor | | | |
|---|---|---|---|
| | SVM | NN1 | NN2 |
| W 90 + PCA 80 + GMM 32 | 41.06% | 52.96% | 53.88% |
| W 90 + PCA 80 + GMM 64 | 39.89% | 49.16% | 51.23% |
| W 90 + PCA 256 + GMM 32 | 41.34% | 53.12% | 50.02% |
| W 90 + PCA 256 + GMM 64 | 43.61% | 54.88% | 50.25% |
| W 60 + PCA 80 + GMM 32 | 49.37% | 57.09% | 54.57% |
| W 60 + PCA 80 + GMM 64 | 47.17% | 52.60% | 50.93% |
| W 60 + PCA 256 + GMM 32 | 45.62% | **57.14%** | 55.58% |
| W 60 + PCA 256 + GMM 64 | 42.43% | 50.91% | 50.24% |

that heavy camera motion is expected, like the "washing dishes" or "drinking water" activities, because it simulates a compensated motion and it proves to be more appropriate when wearable cameras are used. While the action recognition overall has improved with the use of the MBH descriptor as opposed to HOF, the drawbacks of not incorporating information about the active or passive status of the objects is certainly evident here. Both methods perform badly in activities that the same object classes appear frequently. Specifically, confusion seems to exist between the classes "making tea" and "making coffee" because it almost always involve person interactions with the same object classes. Another similar example is the confusion between the "combing hair", "brushing teeth", and "dental floss" classes that are all taking place inside a bathroom with the same objects being visible from the camera.

TABLE III
COMPARISON WITH SOA ON THE ADL DATASET

| Method | Performance (mAP%) |
|---|---|
| Boost-RSTP [13] | 33.7% |
| Boost-RSTP + OCC [13] | 38.7% |
| Bag-of-objects [15] | 32.7% |
| Bag-of-objects + Active model [15] | 36.9% |
| Cascaded Interactional Network [21] | 55.2% |
| Ours - Bag-of-Micro-Actions with HOF (best) | 52.86% |
| Ours - Bag-of-Micro-Actions with MBH (best) | **57.14%** |

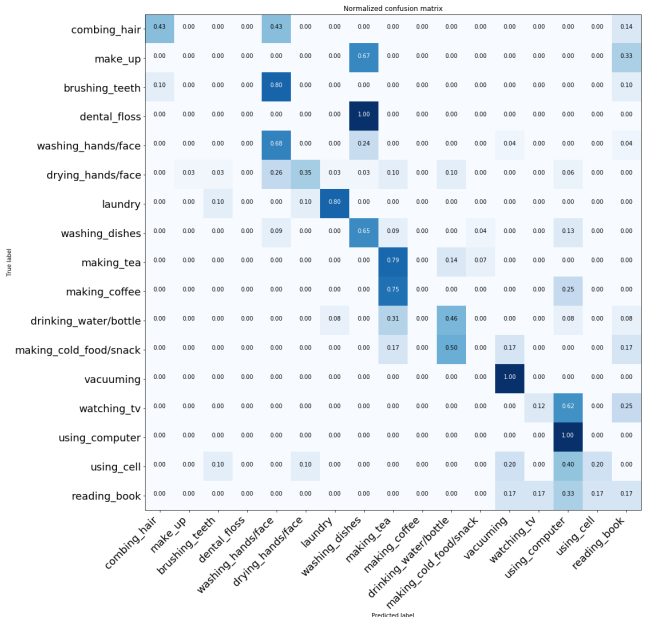


Fig. 3. Confusion matrix of our activity recognition method with HOF descriptors.

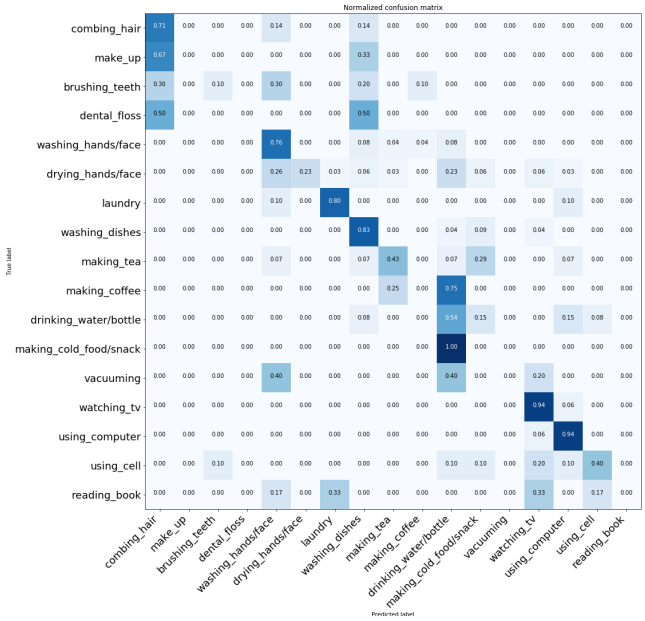


Fig. 4. Confusion matrix of our activity recognition method with MBH descriptors.

## IV. Conclusions

In this paper, we introduced a new approach for activity recognition from wearable cameras by detecting objects and then incorporating their motion patterns into low level micro-action descriptors. We represented activities using a Bag-of-Micro-Actions scheme using GMM clustering and Fisher vector encoding. Our next steps will be to develop an object detection algorithm that discriminates between active and passive objects so as to weight those two classes of objects differently and to leverage hand movements in order to include gesture patterns into the overall framework.

## References


[1] S. Bambach, D. J. Crandall, and C. Yu. Viewpoint integration for hand-based recognition of social interactions from a first-person view. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 351–354. ACM, 2015.

[2] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 1949–1957. IEEE, 2015.

[3] C. F. Crispim-Junior, V. Buso, K. Avgerinakis, G. Meditskos, A. Briassouli, J. Benois-Pineau, I. Y. Kompatsiaris, and F. Bremond. Semantic event fusion of different visual modality concepts for activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1598–1611, 2016.

[4] C. F. Crispim-Junior, A. Gómez Uría, C. Strumia, M. Koperski, A. König, F. Negin, S. Cosar, A. T. Nghiem, D. P. Chau, G. Charpiat, et al. Online recognition of daily activities by color-depth sensing and knowledge models. *Sensors*, 17(7):1528, 2017.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer, 2006.

[6] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 407–414. IEEE, 2011.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[9] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE CVPR*, 2017.

[10] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool. Fast optical flow using dense inverse search. In *European Conference on Computer Vision*, pages 471–488. Springer, 2016.

[11] K. S. Kumar and R. Bhavani. Egocentric activity recognition using histogram oriented features and textural features. 2017.

[12] K. S. Kumar and R. Bhavani. Human activity recognition in egocentric video using pnn, svm, knn and svm+ knn classifiers. *Cluster Computing*, pages 1–10, 2017.

[13] T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *BMVC*, volume 2, page 3, 2013.

[14] G. Meditskos, P.-M. Plans, T. G. Stavropoulos, J. Benois-Pineau, V. Buso, and I. Kompatsiaris. Multi-modal activity recognition from egocentric vision, semantic enrichment and lifelogging applications for the care of dementia. *Journal of Visual Communication and Image Representation*, 51:169–190, 2018.

[15] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012.

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[17] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah. Improved scene identification and object detection on egocentric vision of daily activities. *Computer Vision and Image Understanding*, 156:92–103, 2017.

[18] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[19] X. Wang, L. Gao, J. Song, X. Zhen, N. Sebe, and H. T. Shen. Deep appearance and motion learning for egocentric activity recognition. *Neurocomputing*, 275:438–447, 2018.

[20] Y. Yan, E. Ricci, G. Liu, and N. Sebe. Egocentric daily activity recognition via multitask clustering. *IEEE Transactions on Image Processing*, 24(10):2984–2995, 2015.

[21] Y. Zhou, B. Ni, R. Hong, X. Yang, and Q. Tian. Cascaded interactional targeting network for egocentric video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1904–1913, 2016.