

Article

# An Unsupervised Framework for Online Spatiotemporal Detection of Activities of Daily Living by Hierarchical Activity Models

Farhood Negin <sup>1,2\*</sup> and Francois Bremond <sup>1,3</sup>

<sup>1</sup> INRIA, Sophia Antipolis, 2004 route des Lucioles, 06902, Sophia Antipolis, France,

<sup>2</sup> Institut Pascal, CNRS, UMR 6602, F-63171 Aubiere, France

<sup>3</sup> Université Côte d'Azur, Nice, France

farhood.negin@inria.fr; francois.bremond@inria.fr

\* Correspondence: farhood.negin@inria.fr

Version September 22, 2019 submitted to Sensors

**Abstract:** Automatic detection and analysis of human activities captured by various sensors (e.g. sequence of images captured by RGB camera) play an essential role in various research fields in order to understand the semantic content of a captured scene. The main focus of the earlier studies has been widely on supervised classification problem, where a label is assigned for a given short clip. Nevertheless, in real-world scenarios, such as in Activities of Daily Living (ADL), the challenge is to automatically browse long-term (days and weeks) stream of videos to identify segments with semantics corresponding to the model activities and their temporal boundaries. This paper proposes an unsupervised solution to address this problem by generating hierarchical models that combine global trajectory information with local dynamics of the human body. Global information helps in modeling the spatiotemporal evolution of long-term activities and hence, their spatial and temporal localization. Moreover, the local dynamic information incorporates complex local motion patterns of daily activities into the models. Our proposed method is evaluated using realistic datasets captured from observation rooms in hospitals and nursing homes. The experimental data on a variety of monitoring scenarios in hospital settings reveals how this framework can be exploited to provide timely diagnose and medical interventions for cognitive disorders such as Alzheimer's disease. The obtained results show that our framework is a promising attempt capable of generating activity models without any supervision.

**Keywords:** Activity recognition; Activity of Daily Living; Assisted living; Hierarchical activity models; Unsupervised modeling

## 1. Introduction

Activity detection has been considered as one of the major challenges in computer vision due to its utter importance for several applications including video perception, healthcare, surveillance, etc. For example, if a system could monitor human activities, it could prevent the elderly from missing their medication doses by learning their habitual patterns and daily routines. Unlike regular activities that usually occur in a closely controlled background (e.g. playing soccer), Activities of Daily Living (ADL) usually happen in uncontrolled and disarranged household or office environments, where the background is not a strong cue for recognition. In addition, ADLs are more challenging to detect and recognize because of their unstructured and complex nature that create visually perplexing dynamics. Moreover, each person has his/her own ways to perform various daily tasks resulting in infinite variations of speed and style of performance which accordingly add extra complexity to detection and recognition tasks.

32 From the temporal aspect, detecting ADLs in untrimmed videos is a difficult task since they are  
33 temporally unconstrained and can occur at any time and in an arbitrarily long video (e.g. recordings  
34 of patients in a nursing home for days and weeks). Therefore, in activity detection, we are not only  
35 interested in knowing the type of the activities happening, but also we want to precisely know the  
36 temporal delineation of the activities in a given video (temporal activity localization).

37 Most of the available state-of-the-art approaches deal with this problem through detection by  
38 classification task [1–3]. These methods classify the generated temporal segments either in the form  
39 of sliding windows in multiple scales [4–6] or another external proposal mechanism [7,8]. These  
40 methods infer the occurring activity by exhaustively applying trained activity classifiers at each time  
41 segment. Although they achieved encouraging performances in short actions and small-scale datasets,  
42 these computationally expensive methods can not be conveniently applied to large-scale datasets and  
43 complex activities such as ADLs. These methods are not capable of precisely predicting flexible activity  
44 boundaries. Temporal scale variability of the activities can be dealt with by using multiple-scale sliding  
45 window approaches, however, such methods are computationally expensive. To compensate the high  
46 computational cost of these methods, a class of methods [4,8,9] influenced by advancements in the field  
47 of object detection [10–12] have been developed in which instead of exhaustive scanning, perform a  
48 quick scan to single out candidate activity segments. The sought after activities are more likely to occur  
49 in these segments. In the second step, the activity classifiers are only applied to the candidate segments,  
50 therefore, remarkably reduce the operational cost. Although these methods have shown good results  
51 on activity recognition tasks [13–15], they rarely use context priors in their models. Another drawback  
52 is that instead of learning an end-to-end deep representation, they use off-the-shelf hand-crafted [16]  
53 or deep [17,18] representations independently learned from images. This will result in a poor detection  
54 performance as these representations are not intended and not optimal for localization.

55 Most of the above-mentioned methods are single-layered supervised approaches. In the training  
56 phase of the activities, the labels are fully (supervised) [16,19,20] or partially (weakly supervised)  
57 [21,22] given. In other studies [23,24], the location of the person or the interacted object is known.  
58 Usually the discovery of temporal structure of activities is done by a linear dynamic system [25], a  
59 Hidden Markov Model [26], hierarchical grammars [27–29] or by spatiotemporal representation [30,31].  
60 These methods have shown satisfying performance on well-clipped videos, however, ADLs consist of  
61 many simple actions forming a complex activity. Therefore, representation in supervised approaches  
62 is insufficient to model these activities and a training set of clipped videos for ADL cannot cover all  
63 the variations. In addition, since these methods require manually clipped videos, they can mostly  
64 follow an offline recognition scheme. There also exist unsupervised approaches [32,33] which are  
65 strong in finding meaningful spatiotemporal patterns of motion. However, global motion patterns  
66 are not enough to obtain a precise classification of ADL. For long-term activities, many unsupervised  
67 approaches model global motion patterns and detect abnormal events by finding the trajectories  
68 that do not fit in the pattern [34,35]. Other methods have been applied to traffic surveillance videos  
69 to learn the regular traffic dynamics (e.g. cars passing a crossroad) and detect abnormal patterns  
70 (e.g. a pedestrian crossing the road) [36]. However, modeling only the global motion pattern in  
71 a single-layered architecture cannot capture the complex structure of long-term human activities.  
72 Moreover, a flat architecture focuses on one activity at a time and intrinsically ignores modeling  
73 of sub-activities. Hierarchical modeling, therefore, enables us to model activities considering their  
74 constituents in different resolutions and allows us to combine both global and local information to  
75 achieve a rich representation of activities.

76 In this work, we propose an unsupervised activity detection and recognition framework to model  
77 as well as evaluate daily living activities. Our method provides a comprehensive representation of  
78 activities by modeling both global and body motion of people. It utilizes a trajectory-based method  
79 to detect important regions in the environment by assigning higher priors to the regions with dense  
80 trajectory points. Using the determined scene regions, it creates a sequence of primitive events in order  
81 to localize activities in time and learn the global motion pattern of people. To describe an activity

82 semantically, we can adapt a notion of resolution by dividing an activity into different granularity  
83 levels. This way, the generated models describe multi-resolution layers of activities by capturing their  
84 hierarchical structure and their sub-activities. Hereupon, the system can move among different layers  
85 in the model to retrieve relevant information about the activities. We create the models to uniquely  
86 characterize the activities by deriving relative information and constructing a hierarchical structure.  
87 Additionally, a large variety of hand-crafted and deep features are employed as an implicit hint to  
88 enrich the representation of the activity models and finally perform accurate activity detection. To  
89 summarize, the core contributions of this paper set forth below:

- 90 • an unsupervised framework for scene modeling and activity discovery
- 91 • dynamic length unsupervised temporal segmentation of videos
- 92 • generating Hierarchical Activity Models using multiple spatial layers of abstraction
- 93 • online detection of activities, as the videos are automatically clipped.
- 94 • finally, evaluating daily living activities, particularly in health care and early diagnosis of  
95 cognitive impairments.

96 following these objectives, we conducted extensive experiments on both public and private datasets  
97 and achieved promising results. The rest of the paper is organized as follows: Section 2 presents the  
98 related studies from the literature. Section 3 explains our suggested approach followed by describing  
99 the conducted experiments in section 4. Lastly, Section 5 concludes the paper.

## 100 2. Related Work

101 **Activity recognition:** For the past few decades, activity recognition has been extensively studied in  
102 which most of the proposed methods are supervised approaches based on the hand-crafted perceptive  
103 features [16,17,20–23,37,38]. The linear models [25,26,39,40] gained the most popularity through  
104 modeling action transitions. Later on, more complicated methods modeling activity’s hierarchical and  
105 graphical relations were introduced [28,29,41].

106 Recent re-emergence of deep learning methods has been led to remarkable performances  
107 in various tasks. That success followed by adapting convolutional networks (CNNs) to activity  
108 recognition problem for the first time in [42]. The inclination toward using CNNs in the field, reinforced  
109 by the introduction of two-stream [43] and 3D-CNN [17] architectures to utilize both motion and  
110 appearance features. Most of these methods are segment-based and usually use a simple method  
111 for aggregating the votes of each segment (frame or snippet). There are also other approaches that  
112 investigate long-range temporal relations of activities through temporal pooling [37,44,45]. However,  
113 the main assumptions in these methods are that the given videos are manually clipped and the  
114 activities take place in the entire duration of the videos. Therefore, the temporal localization of those  
115 activities is not taken into account.

116 **Temporal and Spatiotemporal Activity detection:** The goal in activity detection is to find both the  
117 beginning and end of the activities in long-term untrimmed videos. The previous studies in activity  
118 detection were mostly dominated by sliding window approaches where the videos are segmented  
119 by sliding a detection window followed by training classifiers on various feature types [4,6,46–48].  
120 These methods are computationally expensive and produce noisy detection performances especially in  
121 activity boundaries.

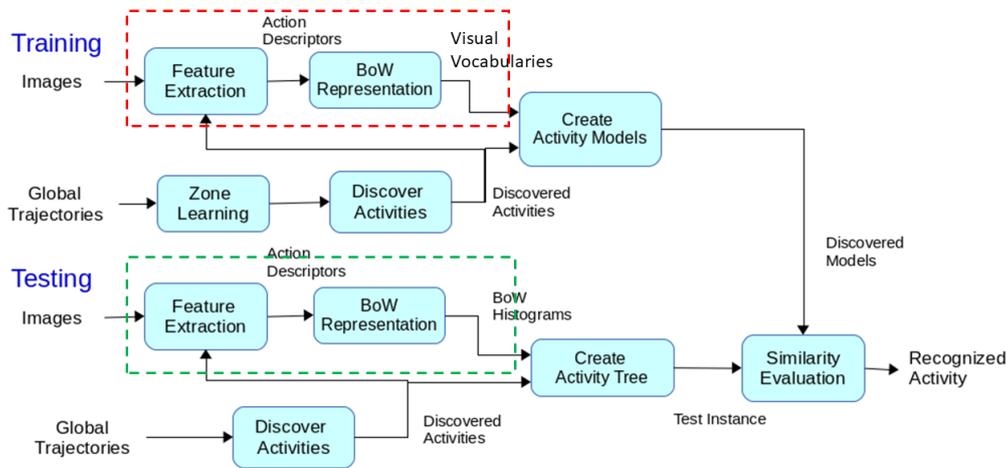
122 Recently, several studies [4,9,49,50] incorporated deep networks and tried to avoid the sliding  
123 window approach and searched for activities with dynamic lengths. This is usually achieved by  
124 temporal modeling of activities using **Recurrent neural network (RNN) or Long short-term memory  
125 (LSTM) networks** [51,52]. For example, [9] uses an LSTM to encode **Convolution3D (C3D)** [17] features  
126 of each segment and classify it without requiring an extra step for producing proposals. Though  
127 their model is still dependant on hand-crafted features. In order to resolve the problem of short  
128 dependencies in RNN based methods, time-series models such as Temporal Convolutional Networks  
129 (TCN) [53,54] employs a combination of temporal convolutional filters and upsampling operations  
130 for acquiring long-range activity relations. However, applying convolutional operations on the

131 local neighborhood for detecting long-range dependencies is not efficient in terms of computational  
132 time. Moreover, many methods use the concept of Actionness [55] to produce initial temporal activity  
133 proposals. Actionness indicates the likelihood of a generic activity localized in the temporal domain.  
134 Reliability of the Actionness hinges upon the correctness of distinguishing the background. Unlike  
135 conventional activity datasets which contain many background segments, long activities in ADL  
136 datasets are usually linked through short background intervals. Accordingly, methods [2,56] relying  
137 on Actionness cannot effectively determine the temporal boundary of ADLs in such datasets.

138 The methods used in [57–61] explore the videos to detect activities in spatial and temporal domains  
139 simultaneously. Some methods [61,62] employ a supervoxel approach to perform spatiotemporal  
140 detection, while others use human detectors [60,63] and treat the detection problem as a tracking  
141 problem [57,59]. Most of these approaches require object detection for a more accurate detection  
142 and therefore, demand exhaustive annotation of objects in long videos which is a tedious and  
143 time-consuming process. Note that the activity detection problem is closely related to object detection  
144 problem from images. A major part of the studies in the literature is inspired by object detection but,  
145 as it is not the focus of this study, we do not review object detection based methods here. However, it  
146 is worth mentioning that although the models currently do not utilize object detection features, yet,  
147 the models have a flexible design which depends on the availability of features, any number and types  
148 of features can be included or excluded from the models.

149 Apart from the supervised methods mentioned above, recently there has been an increasing  
150 attention towards methods with unsupervised learning of activities. A pioneer study conducted  
151 by Guerra-Filho and Aloimonos [64] to overcome the problem of temporal segmentation of human  
152 motion which does not require training data. They suggested a basic segmentation method followed  
153 by clustering step relied on motion data. Based upon these motion descriptors, they made use  
154 of a parallel synchronous grammar system to learn sub-activities of a long activity analogous to  
155 identify words in a complete sentence. Another study performed by Fox et al. [65] made use of  
156 the non-parametric Bayesian approach to model pattern of several related atomic elements of an  
157 activity identical to elements of a time series without any supervision. Similarly, Emonet et al. [66]  
158 proposed an unsupervised Non-parametric Bayesian methods based on Hierarchical Dirichlet Process  
159 (HDP) to discover recurrent temporal patterns of words (Motifs). The method automatically finds  
160 the number of topics, number of time they occur and the time of their occurrence. Furthermore,  
161 several methods took advantage of temporal structure of video data for adjusting parameters of deep  
162 networks without using any labeled data for training [67,68]. Some others [69–72] utilized temporal  
163 pattern of activities in an unsupervised way for representation and hence, for detection of activities.  
164 Lee et al. [71] formulated representation learning as a sequence sorting problem by exploiting the  
165 temporal coherence as a supervisory hint. Temporally shuffled sequence of frames were taken as  
166 input for training a convolutional neural network to determine the correct order of the shuffled  
167 sequences. In another study conducted by Ramanathan et al.[72], a ranking loss based approach was  
168 presented for incorporating temporal context embedding based on past and subsequent frames. A  
169 data augmentation technique was also developed to emphasize the effect of visual diversity of context  
170 embedding. Fernando et al. [70] leveraged the parameters of a frame ranking function as a new video  
171 representation method to encode temporal evolution of activities in the videos. The new representation  
172 provide a latent space for each video where they use a principled learning technique to model activities  
173 without requiring annotation of atomic activity units. Similarly, [73] encoded structured representation  
174 of postures and their temporal evolution as motion descriptors for activities. A combinatorial sequence  
175 matching method is proposed to realize the relationship between frames and a CNN is utilized to  
176 detect the conflict of transitions.

177 So far, state-of-the-art methods are constrained by full supervision and require costly frame level  
178 annotation or at least ordered list of activities in untrimmed videos. By growing the size of video  
179 datasets, it is very important to discover activities in long untrimmed videos. Therefore, recent works  
180 propose unsupervised approaches to tackle the problem of activity detection in untrimmed videos.



**Figure 1.** The flow diagram of the unsupervised framework: Training and Testing phases. The red dashed box shows the training of the visual codebooks of the descriptors. The green box in the testing phase shows the descriptor matching procedure.

181 In this work, we use training videos to specify temporal clusters of segments that contain similar  
 182 semantics throughout all training instances.

### 183 3. Unsupervised Activity Detection Framework

184 The proposed framework provides a complete representation of human activities by incorporating  
 185 (global and local) motion and appearance information. It automatically finds important regions in the  
 186 scene and creates a sequence of primitive events in order to localize activities in time and learn the  
 187 global motion pattern of people. To perform accurate activity recognition, it uses a large variety of  
 188 features such as Histogram of oriented Gradients (HOG), Histogram of optical flow (HOF) or deep  
 189 features as an implicit hint.

190 As figure 1 shows, first, long-term videos are processed to obtain trajectory information of the  
 191 people's movement (input). This information is used to learn scene regions by finding the parts of  
 192 the scene with a higher prior for activities to occur, i.e. dense regions in terms of trajectory points. A  
 193 common approach is to assume that there is only one kind of action occurs inside a region [34,36,74].  
 194 However, in unstructured scene settings, this assumption may not be valid. In order to distinguish  
 195 actions occurring inside the same region, we benefit from the local motion and appearance features  
 196 (visual vocabularies). The learned regions are employed to create primitive events which basically  
 197 determine primitive state transitions between adjacent trajectory points. Based on the acquired  
 198 primitive events, a sequence of discovered (i.e. detected) activities is created to define the global  
 199 motion pattern of people, such as staying inside a region or moving between regions. For each  
 200 discovered activity, motion statistics, such as time duration, etc., are calculated to represent the global  
 201 motion of the person. Finally, a model of a certain activity is constructed through the integration of  
 202 all extracted features and attributes. During the testing phase, the learned regions are used to obtain  
 203 primitive events of the test video. Again, the video is clipped using discovered zones and the action  
 204 descriptors are extracted for each discovered activity. Similar to the training phase, for each discovered  
 205 activity, by combining the local motion information with global motion and other attributes, an activity  
 206 model is constructed. To recognize activities, a comparison is performed between trained activity  
 207 models and acquired test activity. A similarity score between the test instance and trained activity  
 208 models are calculated by comparing global and local motion information of the models. Finally, the  
 209 activity model with the maximum similarity score is considered as recognized activity of the test

210 instance. Through all the steps, an online scheme is followed to perform continuous activity detection  
 211 in assisted living scenarios. The subsequent sections describe different parts of the framework in more  
 212 details.

### 213 3.1. Feature Extraction

214 For local feature detection, improved dense trajectories [75] are employed which densely sample  
 215 points of interests and track them in consecutive frames of a video sequence. **The points of interests are**  
 216 **sampled using a  $W$  pixels sized grid in multiple scales. Each trajectory is track separately at each scale**  
 217 **for  $L$  frames and the trajectories exceeding this limit are removed from the process.** Once the trajectories  
 218 are extracted, the descriptors in the local neighbourhood of interest-points are computed. There are  
 219 three different types of descriptors extracted from the interest-points: **Trajectory shape, motion** (HOF  
 220 and **Motion Boundaries Histogram a.k.a MBH** [75]) and **appearance** (HOG [76]) descriptors.

221 Given a trajectory of length  $L$ , its shape can be described by a sequence ( $S = (\Delta P_t, \dots, \Delta P_{t+L-1})$ )  
 222 of displacement vectors:  $\Delta P = (P_{t+1} - P_t)$ . The final descriptor (trajectory shape descriptor a.k.a TSD)  
 223 is computed by normalizing the magnitude of the displacement vector. Other than spatial scales, the  
 224 trajectories are also calculated in multiple temporal scales in order to represent actions done with  
 225 speed.

226 Motion descriptors (HOF and MBH) are computed in a volume around the detected interest-points  
 227 and throughout their trajectories (spatiotemporal volume). Size of the constructed volume is  $N \times N$   
 228 pixels around the interest-point and  $L$  frames long. For all of the grids in the spatiotemporal volume,  
 229 the descriptors are calculated and concatenated to represent the final descriptor. While motion-based  
 230 descriptors focus on the representation of the local motion, appearance descriptor (HOG) represents  
 231 static appearance information by calculating Gradient vectors around the calculated trajectory point.

232 **Geometrical descriptors** are also used for representing the spatial configuration of the skeleton  
 233 joint information and model human body pose in each frame. To represent the skeleton, both joints'  
 234 Euclidean distances and angles in polar coordinate are calculated using normalized joint positions.  
 235 In order to preserve temporal information in pose representation, a feature extraction scheme based  
 236 on temporal sliding window is adapted [77]. At each time instance, Euclidean distances between  
 237 all the joints are calculated. Besides, for each joint, distance from other instances' joints included in  
 238 the sliding window is calculated and stored. If  $J_i^t$  represents features of joint  $i$  at time  $t$  and  $w$  shows  
 239 the sliding window size:  $J_i^t = [x_i^t, y_i^t]$  defines raw skeleton features at time  $t$ , where  $i = 1, \dots, 8$ . Then,  
 240  $F^d$  calculates the distance descriptor:  $F^d = \sqrt{(x_i^t - x_j^{t'})^2 + (y_i^t - y_j^{t'})^2}$ . Similarly, to calculate angular  
 241 feature in polar coordinate, we use:  $F^a = \arctan(x_i^t - x_j^{t'}, y_i^t - y_j^{t'})$ , where  $t' \in \{t, t-1, \dots, t-w\}, t' > 0$   
 242 and  $i, j = 1, 2, \dots, 8$  for both equations. Combining these features produces the final descriptor vector  
 243  $F = [F^d, F^a]$ .

244 In order to compare the effect of hand-crafted and deep features on our generated activity models,  
 245 the framework also uses **Trajectory-Pooled Deep-Convolutional Descriptors (TDD)** introduced in  
 246 [37]. Computing these features are similar to dense trajectory descriptors. The main difference here  
 247 is that rather than computing the hand-crafted features around the spatiotemporal volume of the  
 248 trajectories, deep features are extracted using convolutional neural network (CNN) maps. To compute  
 249 these features, multi-scale convolutional feature maps pool deep features around the interest-points of  
 250 the detected trajectories. The two-stream ConvNet architecture proposed by Simonyan [43] is adapted  
 251 for TDD feature extraction. The two-stream CNN consists of two separate CNNs: spatial and temporal  
 252 networks. The motion features (temporal) are trained on optical flow and extracted using conv3 and  
 253 conv4 layers of CNN. Additionally, for the training of the appearance features (spatial) on RGB frames,  
 254 conv4 and conv5 layers of CNN are used.

### 255 3.2. Global Tracker

256 Information about the global position of the subjects is indispensable in order to achieve an  
 257 understanding of long-term activities. For person detection, the algorithm in [78] is applied that  
 258 detects head and shoulders from RGBD images. Trajectories of the detected people in the scene are  
 259 obtained using the multi-feature algorithm in [79] using 2D size, 3D displacement, color histogram, the  
 260 dominant color, and covariance descriptors as a feature and the Hungarian algorithm [80] to maximize  
 261 the reliability of the trajectories. We use the control algorithm in [81] to tune tracking parameters in an  
 262 online manner. The output of the tracking algorithm is the input for the framework:

$$Input = \{Seq_1, \dots, Seq_n\} \quad (1)$$

263 where  $Seq_i = Traj_1, \dots, Traj_T$ .  $i$  is the label of the tracked subject and  $T$  is the number of trajectories in  
 264 each sequence. Each scene region characterizes a spatial part of the scene and will be represented as a  
 265 Gaussian distribution:  $SR_i \sim (\mu^i, \sigma^i)$ .

### 266 3.3. Scene Model

267 In most of the trajectory-based activity recognition methods, a priori contextual information is  
 268 ignored while modeling the activities. The proposed framework performs automatic learning of the  
 269 meaningful scene regions (topologies) by taking into account the subject trajectories. The regions are  
 270 learned at multiple resolutions. By tailoring topologies at different levels of resolution, a hierarchical  
 271 scene model is created. A topology at level  $l$  is defined as a set of scene regions (SR):

$$T_{level_l} = \{SR_0, \dots, SR_{k-1}\} \quad (2)$$

272  $k$  indicates the number of scene regions defining the resolution of the topology. The scene regions  
 273 are obtained through clustering which takes place in two stages. This two stages clustering helps  
 274 to reduce the effect of outlier trajectory points in the overall structure of the topologies. In the **first**  
 275 **stage**, the interesting regions for each subject in the training set are found by clustering their trajectory  
 276 points. For each  $Seq$ , the clustering algorithm produces  $k$  clusters:  $Cluster(Seq_i) = \{Cl_1, \dots, Cl_k\}$  where  
 277 each resulted cluster characterizes the scene based on the motion information of subject  $i$ .  $\mu$  and  
 278  $\omega$  parameters of the distribution of the  $SR_i$  are calculated from the clustering.  $C^{th}$  cluster center  
 279 ( $Cl_c$ ) corresponds to scene region  $i$  ( $SR_i$ ). For  $SR_i$ ,  $\mu$  is the spatial coordinate of the cluster centroid:  
 280  $SR_i(\mu) = centroid(Cl_c)$  and the standard deviation  $\sigma$  is computed from the point coordinate sequence  
 281 of the trajectory set. The **second stage** of the clustering merges individual scene regions into a single  
 282 comprehensive set of regions. Each region is a new cluster ( $Cl$ ) in the second stage partitioning the  
 283 obtained cluster centroids in the first stage. K-means algorithm is used for the clustering where the  
 284 optimal value of  $K$  is calculated based on the Bayesian Information Criterion (BIC) [82]. We define a  
 285 scene model as a set of scene regions (topologies) at different resolutions:

$$SceneModel = \langle Topology_{highlevel}, Topology_{midlevel}, Topology_{lowlevel} \rangle \quad (3)$$

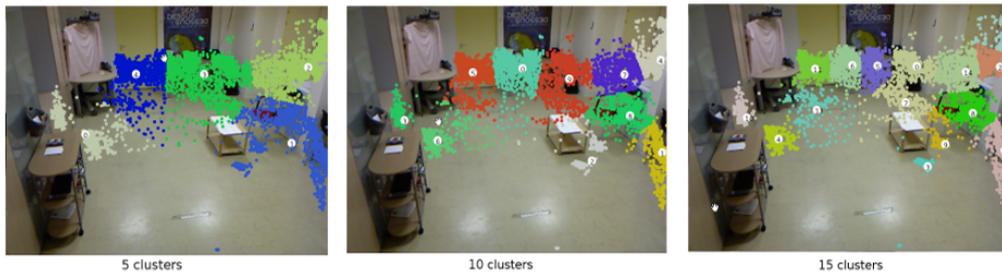
286 We create a model with topologies at three levels, each aims to describe the scene at a high, medium  
 287 and low degree of abstraction. Figure 2 depicts an example of the calculated scene regions in a hospital  
 288 room in CHU dataset<sup>1</sup>.

### 289 3.4. Primitive Events

290 To fill the gap between the low-level image features and high-level semantic description of the  
 291 scene, an intermediate block capable of linking the two is required. Here, we describe a method that

---

<sup>1</sup> <https://team.inria.fr/stars/demcare-chu-dataset/>



**Figure 2.** Example of k-means clustering using city-block distance measurements of CHU Nice dataset. The number of clusters is set to 5, 10 and 15.

292 defines a construction block for learning the activity models. With a deeper look at the activity  
 293 generation process, it can be inferred that the abstraction of low-level features into high-level  
 294 descriptions does not happen in a single step and this transition is gradual. As a solution, we  
 295 use an intermediate representation named Primitive Event (PE). Given the two consecutive trajectory  
 296 data points ( $Traj_i$  and  $Traj_j$ ), by using their distance from the cluster centroids, their corresponding  
 297 scene regions (StartRegion and EndRegion) can be found. A primitive event is represented as a pair of  
 298 directed scene regions of these trajectory points:

$$PrimitiveEvent = (StartRegion \rightarrow EndRegion) \quad (4)$$

299 where  $StartRegion$  and  $EndRegion$  variables take values of SR indices. For example, if  $StartRegion$  of  
 300  $Traj_i$ :  $SR_2$  and  $EndRegion$  of  $Traj_j$ :  $SR_4$  then, we will have  $(2 \rightarrow 4)$  as a primitive event. PE describes an  
 301 atomic motion block and is used for characterizing motion of a person in a scene. This way, a whole  
 302 sequence of trajectory can be translated into PEs. A *Primitive Event*'s type is *Stay*, when the region  
 303 labels (Such as  $SR_1$ ) stay constant between two time intervals. It is equivalent to a sequence of *stays* in  
 304 the scene region  $P$ :

$$Primitive Event = Stay_{P,P} \quad (5)$$

305 When a *Primitive Event*'s type is *Change*, a change of region (from region  $P$  to region  $Q$ ) between two  
 306 successive time instants (i.e. two successive trajectory points) occurs. It is equivalent to a region  
 307 transition:

$$Primitive Event = Change_{P,Q} \quad (6)$$

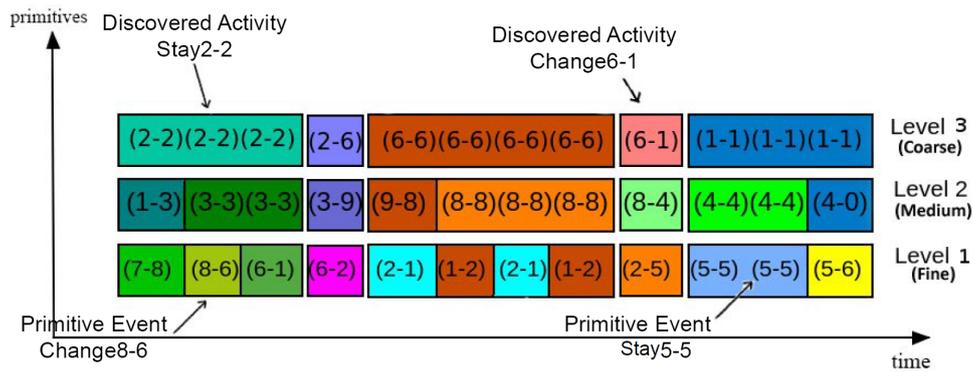
308 The duration of the current status (stay/change) can be calculated simply by  $Duration =$   
 309  $\frac{EndEventFrame - BeginEventFrame}{fps}$  where  $fps$  is the frame rate of the recorded images. Using a learned  
 310 topology  $T$  for every video sequence, a corresponding primitive event sequence  $PE_{seq}$  is calculated:

$$PE_{seq} = (\langle PE_1, \dots, PE_n \rangle, T) \quad (7)$$

311 A primitive Event sequence provides information regarding the underlying structure of long-term  
 312 activities.

### 313 3.5. Activity Discovery (detection)

314 We refer to the detection of the boundaries of the activities as *Activity Discovery*. Annotating the  
 315 beginning and end of the activities is a challenging task even for humans. The start/end time of the  
 316 annotated activities varies from one human annotator to another. The problem is that humans tend to  
 317 pay attention to one resolution at a time. For example, when a person is sitting on a chair, the annotated



**Figure 3.** A sample video encoded with primitive events and discovered activities in three resolution levels.

318 label is "sitting". Later, when the subject "moves an arm", she is still sitting. Discovering activities using  
 319 a different resolution of the trained typologies helps to automatically detect these activity parts and  
 320 sub-parts at different levels of activity hierarchy using previously created semantic blocks (Primitive  
 321 Events). Input for activity discovery process is a spatiotemporal sequence of activities described  
 322 by primitive events. After the activity discovery process: 1) The beginning and end of all activities  
 323 in a video are estimated and the video is automatically clipped. 2) The video is classified naively  
 324 into discovered activities indicating similar activities in the timeline. A discovered activity (DA) is  
 325 considered either as 1) staying in current state ("Stay") or 2) changing of the current state ("Change").  
 326 Basically, a *Stay* pattern is an activity that occurs inside a single scene region and is composed of  
 327 primitive events with the same type:

$$Discovered\ Activity = Stay_{P \rightarrow P} = \{Stay\ PE_s\} \quad (8)$$

A "Change" pattern is an activity that happens between two topology regions. A "Change" activity consists of a single primitive event of the same type:

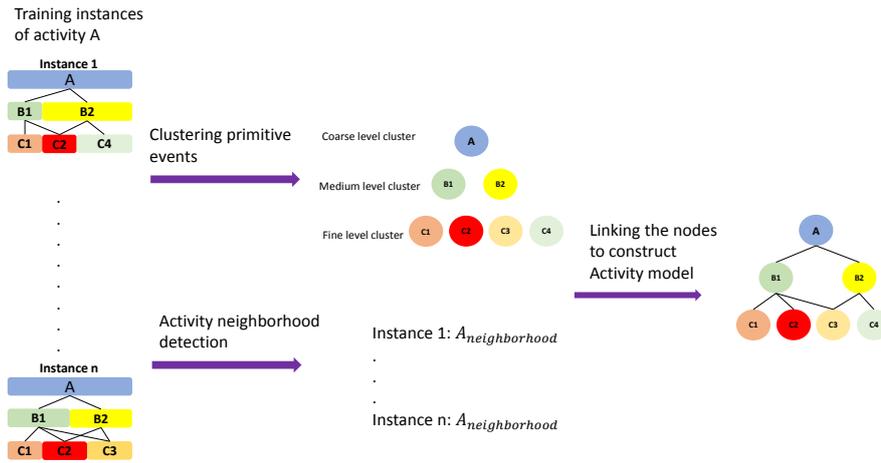
$$Discovered\ Activity = Change_{P \rightarrow Q} = Change\ PE \quad (9)$$

328 Although detection of primitive events takes place at three different resolutions, the activity discovery  
 329 process only considers the coarse resolution. Therefore, after discovery process, the output of the  
 330 algorithm for the input sequence is a data structure containing information about the segmented input  
 331 sequence in the coarse level and its primitive events in two other lower levels. This data structure  
 332 holds spatiotemporal information similar to the structure in Figure 3. The algorithm for this process  
 333 simply checks for primitives' boundaries and constructs the data structure for each discovered activity.  
 334 Employing DAs and PEs, it shows the hierarchical structure of an activity and its sub-activities.

335 Although *Discovered Activities* present global information about the movement of people, it is not  
 336 sufficient to distinguish activities occurring in the same region. Thus, for each discovered activity, body  
 337 motion information is incorporated by extracting motion descriptors (section 3.1). These descriptors  
 338 are extracted in a volume of  $N \times N$  pixels and  $L$  frames from videos. Fisher Vector (FV) method [83] is  
 339 then followed to obtain a discriminative representation of activities. The descriptors are extracted for  
 340 all *Discovered Activities* that are automatically computed. The local descriptor information is extracted  
 341 only for *Discovered Activities* at the coarse resolution level.

### 342 3.6. Activity Modeling

343 Here, the goal is to create activity models with high discriminative strength and less susceptibility  
 344 to noise. We use attributes of an activity and its sub-activities for modeling and accordingly, learning  
 345 is performed automatically using the DAs and PEs in different resolutions. Learning such models



**Figure 4.** The process of creating activity tree. The PEs from the training instances are clustered into nodes and at the same time, the neighborhood set is detected. The final structure is constructed with those building blocks.

346 enables the algorithm to measure the similarity between them. To create the models, a method for  
 347 assembling the DAs and PEs from different resolutions is required. This is achieved by the concept of  
 348 hierarchical neighborhood.

### 349 3.6.1. Hierarchical Neighborhood

350 The hierarchical representation of activity  $A$  at resolution level  $l$  is a recursive representation of  
 351 the links between  $A$  and its primitive events  $B_i$  at the finer resolutions:

$$A_{neighborhood} = ((B1, B1_{neighborhood}), \dots, (Bn, Bn_{neighborhood})) \quad (10)$$

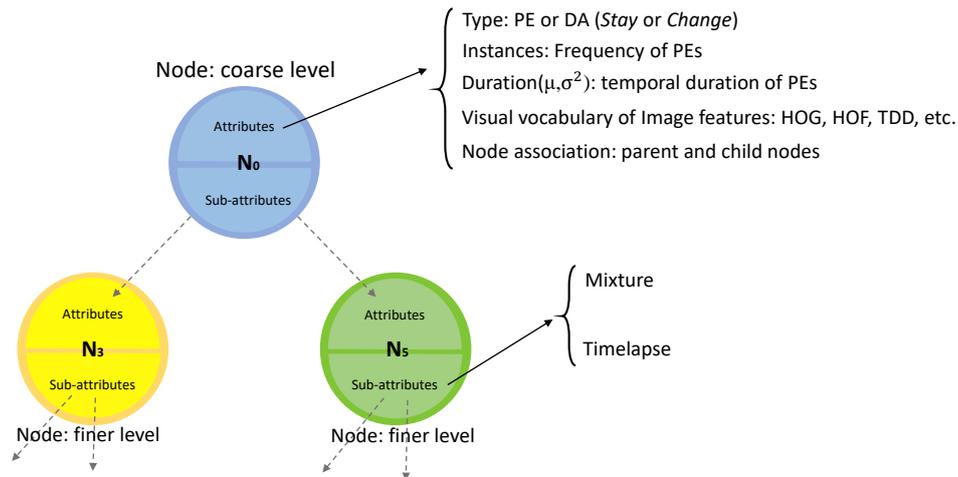
352  $B1, \dots, Bn$  are the primitive events of  $A$  in the next finer resolution. The links between the different  
 353 levels are established using temporal overlap information. For example, primitive event  $B$  is  
 354 sub-activity of activity  $A$  in a higher level if their temporal interval overlaps in the activity timeline.  
 355 Formally,  $B$  is sub-activity of  $A$  if the following statement holds:

$$\begin{aligned} & ((startFrame_A \leq startFrame_B) \wedge (endFrame_A \geq startFrame_B)) \\ & \parallel ((startFrame_A \leq endFrame_B) \wedge (endFrame_A \geq endFrame_B)) \\ & \parallel ((startFrame_A \leq startFrame_B) \wedge (endFrame_A \geq endFrame_B)) \\ & \parallel ((startFrame_A \geq startFrame_B) \wedge (endFrame_A \leq endFrame_B)) \end{aligned} \quad (11)$$

356 By applying 10 to a discovered activity, we can find the primitives in its neighborhood. This automatic  
 357 retrieval and representation of the neighborhood of a DA help in creating the hierarchical activity  
 358 models.

### 359 3.6.2. Hierarchical Activity Models

360 Hierarchical activity model (HAM) is defined as a tree that captures the hierarchical structure  
 361 of daily living activities by taking advantage of the hierarchical neighborhoods to associate different  
 362 levels. For an input DA ( $A_{neighbourhood}$ ) and its neighborhood, the goal is to group similar PEs obtained  
 363 by clustering to create nodes ( $N$ ) of the activity tree. Clustering is performed using *Type* attribute of  
 364 the PEs which groups PEs of the same type in one cluster. This process is repeated for all levels. After



**Figure 5.** An example of model architecture in node level where each node is composed of attributes and sub-attributes.

365 clustering, nodes of the tree model are determined followed by linking them together to construct the  
 366 hierarchical model of the tree. The links between the nodes are realized from the activity neighborhood  
 367 of each node (Figure 4 shows the complete procedure of creating an activity tree from neighborhood  
 368 set instances of a DA). After linking, a complete tree structure of the given DA is obtained and the  
 369 model is completed by adding attribute information for nodes of the tree. Each node in the activity  
 370 tree contains information about the similar detected primitive events sharing similar properties such  
 371 as duration and type of the primitive as well as similar sub-activities in the lower level. So, a node  
 372 is the representative of all the similar primitives in that level. Each node has two types of properties.  
 373 The node attributes that store information about primitive events such as average duration of its  
 374 constituents as well as information about parent node and the associated nodes in the lower level of  
 375 the hierarchy. The nodes can keep different spatial and temporal attributes about the activity and its  
 376 sub-activities. The former type is consisted of:

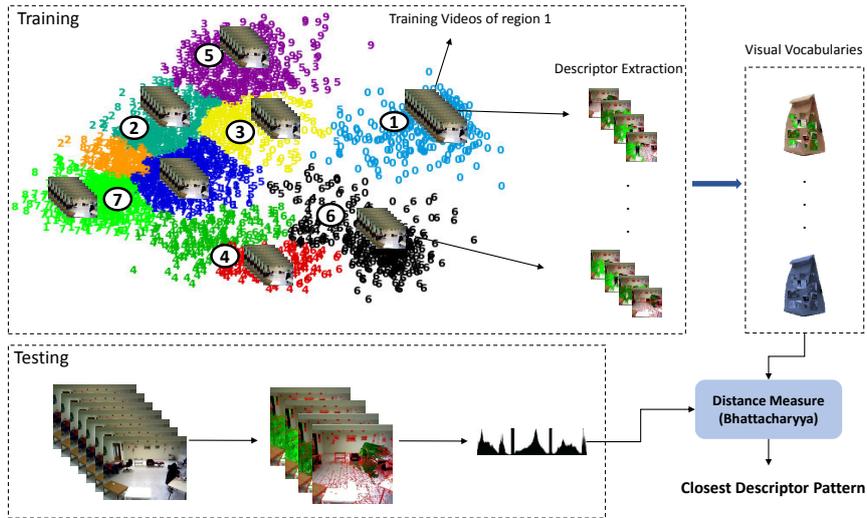
- 377 • *Type attribute* is extracted from the underlying primitive or discovered activity (in case of the root  
 378 node). For node  $N$ ,  $Type_N = Type_{PE}$  or  $Type_{DA}$ , where  $Type$  of PEs and DAs are either *Stay* or  
 379 *Change* states.
- 380 • *Instances* list PEs of training instances indicating the frequency of each PE included in the node.
- 381 • *Duration* is a Gaussian distribution  $Duration(\mu_d, \sigma_d^2)$  describing the temporal duration of the PEs  
 382 ( $\{PE_1, PE_2, \dots, PE_n\}$ ) or discovered activities ( $\{DA_1, DA_2, \dots, DA_n\}$ ) of the node. It is frame  
 383 length of the primitives or discovered activities calculated as:

$$\mu_d = \sum_{i,j=1}^n \frac{(endframe_{PE_i,orDA_j} - startframe_{PE_i,orDA_j})}{n} \quad (12)$$

$$\sigma_d^2 = E[(endframe_{PE_i,orDA_j} - startframe_{PE_i,orDA_j} - \mu_d)^2] \quad (13)$$

384 where  $n$  is the number of PEs or DAs.

- 385 • *Image Features* store different features extracted from the discovered activities. There is no  
 386 limitation on the type of feature. It can be extracted hand-crafted features, geometrical or deep  
 387 features (section 3.1). It is calculated as the histogram of the features of the instances in the  
 388 training set.
- 389 • *Node association* indicates the parent node of the current node (if it is not the root node) and the  
 390 list of neighborhood nodes in the lower levels.



**Figure 6.** The process of learning visual codebook for each activity model and matching the given activity's features with the most similar dictionary: Training and Testing phases.

391 The above-mentioned attributes do not describe the relationship between the nodes which is important  
 392 in the overall description of the activities. In order to model the relationship among the nodes, for  
 393 each node, two other attributes are defined regarding their sub-nodes: *Mixture* and *Timelapse*. *Mixture*  
 394 shows contribution of the type of the sub-activities ( $Stay_{2-2}$ ) in the total composition of sub-nodes.  
 395 This number is modeled with a Gaussian mixture  $\Theta_{type}^{mixture}$ . *Timelapse* of the nodes (with the same type  
 396 and level in different training instances) represents the distribution of the temporal duration of the  
 397 sub-nodes. This attribute is also computed as a Gaussian distribution  $\Theta_{type}^{timelapse}$ . The created HAM  
 398 structure is a hierarchical tree that provides recursive capabilities. Accordingly, it makes the calculation  
 399 of the attributes and the score in the recognition step efficient and recursive. **Figure 5 illustrates an**  
 400 **example of a HAM model with its nodes and their attributes and sub-attributes.**

### 401 3.7. Descriptor Matching of Tree Nodes

402 Descriptor matching can be denoted as a method that captures the similarity between a given  
 403 local dynamic information of an activity and a set of calculated multi-dimensional distributions. The  
 404 obtained descriptor vectors ( $H$ ) characterize local motion and appearance of a subject. Knowing the  
 405 vector representation of the descriptors of discovered activities enables the use of a distance (Eq. 14)  
 406 measurement to characterize the similarity between different activities. As it is shown in figure 6, in  
 407 training, the scene model is used to clip the long videos to the short clips belonging to each region.  
 408 Next, the descriptors of the clipped videos are extracted and employed to learn a visual codebook  
 409  $V$  (one for each region) by clustering the descriptors (Using k-means). The codebook of each region  
 410 is stored in the created activity model of that region. During the testing phase, when a new video is  
 411 detected by the scene model, its descriptors are extracted and the feature vectors are created. These  
 412 feature vectors are encoded with the learned dictionaries of the models. The distance of the current  
 413 descriptor is calculated with the trained codebooks of all regions (to find the closest one) using the  
 414 Bhattacharyya distance:

$$Distance(H, V) = \sum_{i=1}^N BC(H, V_i) \quad (14)$$

415 where  $N$  is the number of learned code words and  $BC$  is the Bhattacharyya coefficient:

$$BC = \sum_{x,y=1}^{N,M} H(x)V_i(y) \quad (15)$$

416  $N$  and  $M$  display dimensions of the descriptor and trained codebooks, respectively. The most  
 417 similar codebook is determined by the minimum distance score acquired. That codebook (and its  
 418 corresponding activity model) is assigned by a higher score in the calculation of the final similarity  
 419 score with the test instance in the recognition phase.

### 420 3.8. Model Matching for Recognition:

421 To measure the similarity among the trained HAM models, different criteria can be considered.  
 422 The assumed criterion can vary from one application to another. While one application can emphasize  
 423 more on the duration of activities, local motion can be more important for others. Although these  
 424 criteria can be set depending on the application, the weights of the feature types are learned to  
 425 determine the importance of each type. The recognition is carried out in five steps as follows:

- 426 1. Perceptual information, such as trajectories of a new subject, is retrieved.
- 427 2. Using the previously learned scene model, the primitive events for the new video are calculated.
- 428 3. By means of retrieved primitive events, the discovered activities are calculated.
- 429 4. Using the collected attribute information, a test instance HAM ( $\omega^*$ ) is created.
- 430 5. The similarity score of the created HAM and trained HAM models are calculated and the activity  
 431 with the highest score is selected as the target activity.

432 Once the activity models are trained, to find the one that matches with an activity in a test video, we  
 433 follow a Bayesian scheme. We choose the final label using the Maximum A Posteriori (MAP) decision  
 434 rule. If  $\Omega = \{\omega_1, \dots, \omega_S\}$ , where  $S = |\Omega|$  represent the set of generated activity models and given the  
 435 data for an observed test video,  $\omega^*$ , we select the activity model,  $\omega_i$ , that maximizes the likelihood  
 436 function [Eq. 16]:

$$p(\omega^*|\omega_i) = \frac{p(\omega^*) p(\omega_i|\omega^*)}{p(\omega_i)} \quad (16)$$

437 where  $p(\omega_i|\omega^*)$  denotes the likelihood function defined for activity models  $\omega_1, \dots, \omega_S$  in model set  $\Omega$ .  
 438 We assume that the activity models are independent. Therefore, *a priori* probability of trained models  
 439  $p(\omega_1, \dots, \omega_S)$  is considered equal. We can eliminate  $p(\omega_i)$  and use the following formula [Eq. 17]

$$\tilde{p}(\omega^*|\omega_i) = p(\omega^*) \prod_{i=1}^S p(\omega_i|\omega^*) \quad (17)$$

440  $p(\omega^*)$  is the relative frequency of  $\omega^*$  in the training set. Since the generated models are constructed  
 441 following a tree structure, the likelihood value should be calculated recursively to cover all nodes of  
 442 the tree. For each model, the recursive probability value is therefore calculated as Eq. 18

$$p(\omega_i|\omega^*) = p(\omega_i^{[l]}|\omega^{*[l]}) + \text{Recur}([l] - 1) \quad (18)$$

443 *Recur* recursively calculates the probabilities of the nodes in lower levels and stops when there  
 444 is no more leaf to be compared. Superscripts index the levels of the tree ( $[l]=1,2,3$ ).  $p(\omega_i^{[l]}|\omega^{*[l]})$   
 445 calculates probability in the current node given  $\omega^*$  and  $p(\omega_i^{[l]}|\omega^{*[l-1]})$  returns the probability values  
 446 of this node's child nodes (sub-activities). Given the data for node  $n$  of the activity in the test video,  
 447  $\omega^*(n) = \{type^*(n), duration^*(n), l^*(n)\}$  and the activity model  $i$ ,  $\omega_i(n) = \{type^i(n), \Delta_{duration}^i(n),$   
 448  $Distance^i(n)\}$ , where  $\Delta_{duration}^i = \{\mu^i, \sigma^i\}$ . The likelihood function for node  $n$  is defined as Eq. 19.

$$\begin{aligned} \tilde{p}(\omega_i(n)^l|\omega^*(n)) &= p(\omega^*(n)|type^* = type^i(n)) * \\ & p(duration^*(n)|\Delta_{duration}^i(n)) * \\ & p(\omega^*(n)|l^* = Distance^i(n)) \end{aligned} \quad (19)$$

$p(\omega^*(n)|type^* = type^i(n))$  checks whether the types of nodes in test tree and trained model are the same or not:

$$p(\omega^*(n)|type = type^i(n)) = \begin{cases} 1 & \text{if } type^* = type^i(n) \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

449  $p(duration^*(n)|\Delta_{duration}^i(n))$  measures the difference between activity instance  $\omega^*$ 's duration and  
450 activity model  $i$  bounded between 0 and 1.

$$p(\omega_*(n)|\mu = \mu_{duration}^i(n)) \propto \exp^{-Dist_{duration}(n)} \quad (21)$$

where

$$Dist_{duration}(n) = \frac{|duration^*(n) - \mu_{duration}^i(n)|}{\sigma^i}$$

451  $p(\omega^*(n)|l = Distance^i(n))$  compares the distance of training node's trained codebooks  $V$  and the test  
452 node's computed descriptor histogram  $H$ .

$$p(\omega^*(n)|l = Distance^i(n)) = \begin{cases} 1 & \text{if } Distance(H, V)^*(n) = \min(Distance^i(n)) \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

It should be noted that the *Distance* information is only available at root level  $l = 0$  (only for DAs). The recursion stops when it traverses all the leaves (exact inference). Once we computed  $p(\omega^*|\Omega)$  for all model assignments, using MAP estimation, the activity model  $i$  that maximizes the likelihood function  $p(\omega_i|\omega^*)$  votes for the final recognized activity label [Eq.23].

$$\hat{i} = \arg \max_i \tilde{p}(\omega^*|\omega_i) \quad (23)$$

## 453 4. Experiments and Discussion

### 454 4.1. Datasets

455 The performance of the proposed framework is evaluated on two public and one private daily  
456 living activity datasets.

#### 457 4.1.1. GAADR Dataset

458 The GAADR [84] activity dataset consists of 25 people with dementia and mild cognitive  
459 impairment who perform ADLs in an environment similar to a nursing home. The GAADR dataset is  
460 public and was recorded under the EU FP7 Dem@Care Project<sup>2</sup> in a clinic in Thessaloniki, Greece. The  
461 camera monitors a whole room where a person performs directed ADLs. The observed ADLs include:  
462 "Answer the Phone", "Establish Account Balance", "Prepare Drink", "Prepare Drug Box", "Water Plant",  
463 "Read Article", "Turn On Radio". A sample of images for each activity is presented in Figure 7 (top  
464 row). Each person is recorded using an RGBD camera of  $640 \times 480$  pixels of resolution. Each video lasts  
465 approximately 10-15 minutes. We randomly selected 2/3 of the videos for training and the remaining  
466 for testing.

#### 467 4.1.2. CHU Dataset

468 This dataset is recorded in the Centre Hospitalier Universitaire de Nice (CHU) in Nice, France.  
469 It contains videos from patients performing everyday activities in a hospital observation room.  
470 The activities recorded for this dataset are "Prepare Drink", "Answer the Phone", "Reading Article",

---

<sup>2</sup> <http://www.demcare.eu/results/datasets>



**Figure 7.** Instances of daily activities provided in GAADR (figures a-d), CHU (figures e-h) and DAHLIA (figures i-l) datasets.

471 "Watering Plant", "Prepare Drug Box" and "Checking Bus Map". A sample of images for each activity  
 472 is illustrated in Figure 7 (middle row). Each person is recorded using an RGBD Kinect camera with  
 473  $640 \times 480$  pixels of resolution, mounted on the top corner of the room. The hospital dataset is recorded  
 474 under the EU FP7 DemCare project<sup>3</sup> and it contains 27 videos. For each person, the video recording  
 475 lasts approximately 15 minutes. Domain experts annotated each video regarding the ADLs. Similar to  
 476 GAADR, for this dataset, we randomly chose 2/3 of the videos for training and the rest for testing.

#### 477 4.1.3. DAHLIA Dataset

478 The DAHLIA dataset [85] consists of a total of 153 long-term videos of daily living activities (51  
 479 videos recorded from 3 different views) from 44 people. The average duration of the videos is 39  
 480 minutes containing 7 different actions (and a Neutral class). The considered ADLs are: "Cooking",  
 481 "Laying Table", "Eating", "Clearing Table", "Washing Dishes", "Housework" and "Working" (figure 7  
 482 bottom row). To evaluate this dataset, we followed a cross-subject protocol in order to compare our  
 483 results with existing literature.

#### 484 4.2. Evaluation Metrics

485 We use various evaluation metrics on each dataset to evaluate our results and compare it with other  
 486 approaches. For the GAADR and CHU datasets, we use Precision and recall metrics. True Positive  
 487 Rate (TPR) or recall is the proportion of actual positives which are identified correctly:  $TPR = \frac{TP}{TP+FN}$ .  
 488 The higher the value of this metric, the better is the performance. Similarly, Positive Predictive Value  
 489 (PPV) or precision is defined as:  $PPV = \frac{TP}{TP+FP}$ . We also use F-score in our comparisons. The detected  
 490 intervals are compared against the ground-truth intervals and an overlap higher than 80% of the  
 491 ground-truth interval is considered as a True Positive detection of that activity.

492 For evaluation of the unsupervised framework, as the recognized activities are not labeled, there  
 493 is no matching ground-truth activity label for them. The recognized activities are labeled such as

<sup>3</sup> <https://team.inria.fr/stars/demcare-chu-dataset/>

	32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score												
Angle	57.6	33.2	0.42	61.2	36.1	0.45	46.9	30.2	0.36	28.1	22.4	0.24	26.7	19.8	0.22
Distance	12.9	9.7	0.11	18.2	14.9	0.16	20.7	16.1	0.18	14.7	12.1	0.13	14.7	15.2	0.14
HOG	81.4	75.2	0.78	84.7	79.6	0.825	77.5	74.3	0.75	82.7	77.6	0.80	84.7	79.8	0.82
HOF	64.6	61.9	0.63	64.9	67.7	0.66	66.1	68.1	0.67	65.4	67.9	0.66	57.4	62.1	0.59
MBHX	71.3	77.2	0.74	74.8	78.2	0.76	79.8	76.1	0.77	67.6	72.1	0.69	69.4	72.8	0.71
MBHY	71.5	68.4	0.69	78.8	76.1	0.77	82.7	84.9	0.83	83.1	85.7	0.84	80.2	79.4	0.79
TDD Spatial	74.5	72.9	0.73	72.8	71.2	0.71	77.5	74.3	0.75	77.5	76.9	0.77	76.4	73.5	0.74
TDD Temporal	73.4	69.1	0.71	73.9	70.6	0.72	72.5	69.9	0.71	79.4	76.2	0.77	81.9	76.9	0.79

**Table 1.** Results related to the unsupervised framework with different feature types on GAADR dataset.

494 "Activity 2 in Zone 1". In order to evaluate the recognition performance, first, we map the recognized  
495 activity intervals on the labeled ground-truth ranges. Next, we evaluate the one-to-one correspondence  
496 between a recognized activity and a ground-truth label. For example, we check which ground-truth  
497 activity label co-occurs the most with "Activity 2 in Zone 1". We observe that in 80% of the time, this  
498 activity coincides with "Prepare Drink" label in the ground-truth. We, therefore, infer that "Activity 2  
499 in Zone 1" represents "Prepare Drink" activity. For this purpose, we create a correspondence matrix for  
500 each activity which is defined as a square matrix where its rows are the recognized activities and the  
501 columns are ground-truth labels. Each element of the matrix shows the number of co-occurrences of  
502 that recognized activity with the related ground-truth label in that column:

$$COR(RA, GT) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

503  $a_{ij} \in \mathbb{Z}^+$  shows the correspondence between activity instance  $i$  and ground-truth label  $j$ .  $RA$  is the  
504 set of recognized activity instances and  $GT$  shows the set of ground-truth labels. We evaluate the  
505 performance of the framework based on the inferred labels. These labels are used for calculating the  
506 *Precision*, *Recall* and *F-Score* metrics.

507 In order to evaluate the DAHLIA dataset, we use metrics based on frame level accuracy. For each  
508 class  $c$  in the dataset, we assume  $TP^c$ ,  $FP^c$ ,  $TN^c$  and  $FN^c$  as the number of True Positive, False Positive,  
509 True Negative and False Negative frames, respectively. Therefore, Frame-wise accuracy is defined  
510 as:  $FA_1 = \frac{\sum_{c \in C} TP^c}{\sum_{c \in C} N_c}$  where  $N_c$  is the number of correctly labeled frames compared to the ground-truth.  
511 F-Score is defined as:  $F - Score = \frac{2}{|C|} \sum_{c \in C} \frac{P^c \times R^c}{P^c + R^c}$  where  $P^c$  and  $R^c$  are precision and recall metrics of  
512 class  $c$ , respectively. We also define Intersection over Union (IoU) metric as:

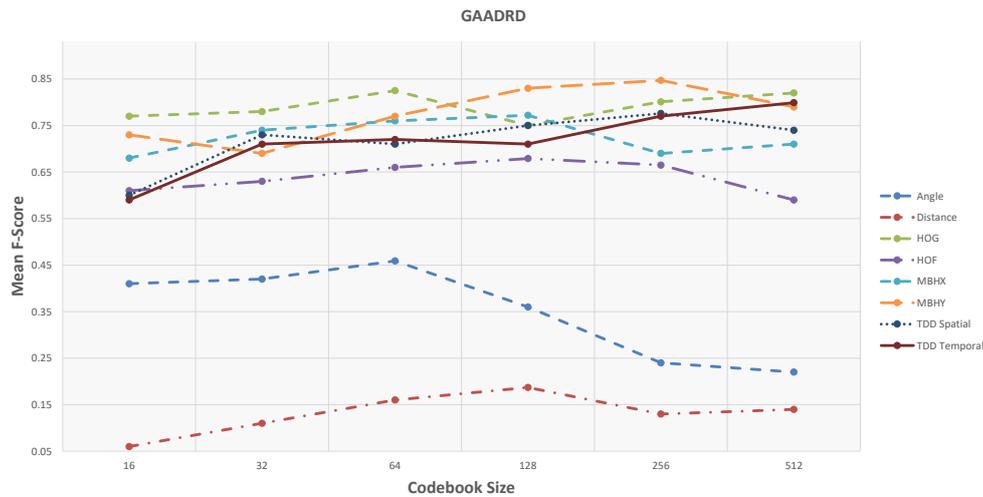
$$IoU = \frac{1}{|C|} \sum_{c \in C} \frac{TP^c}{TP^c + FP^c + FN^c} \quad (24)$$

513  $C$  is the total number of action classes.

### 514 4.3. Results and Discussion

515 First, the results and evaluations of the three datasets are reported and then compared with  
516 state-of-the-art methods. Different codebook sizes are examined for the Fisher vector dictionaries: 16,  
517 32, 64, 128, 256 and 512. Table 1 and figure 8 show the accuracy of activity detection based on Precision  
518 and Recall metrics using the feature type with the highest accuracy. In the case of GAADR dataset,  
519 the best result achieved with incorporated **Motion Boundaries Histogram in Y axis (MBHY)** descriptor  
520 in the activity models with codebook size set to 256.

521 Based on the obtained results, there is no special trend regarding the codebook size. For some  
522 features (MBHY and TDD spatial), the performance increases with an increase in the codebook size and  
523 drops when the codebook size becomes much bigger. For TDD temporal feature, performance increases  
524 linearly with the codebook size. For the geometrical features, particularly for the Angle feature, there is



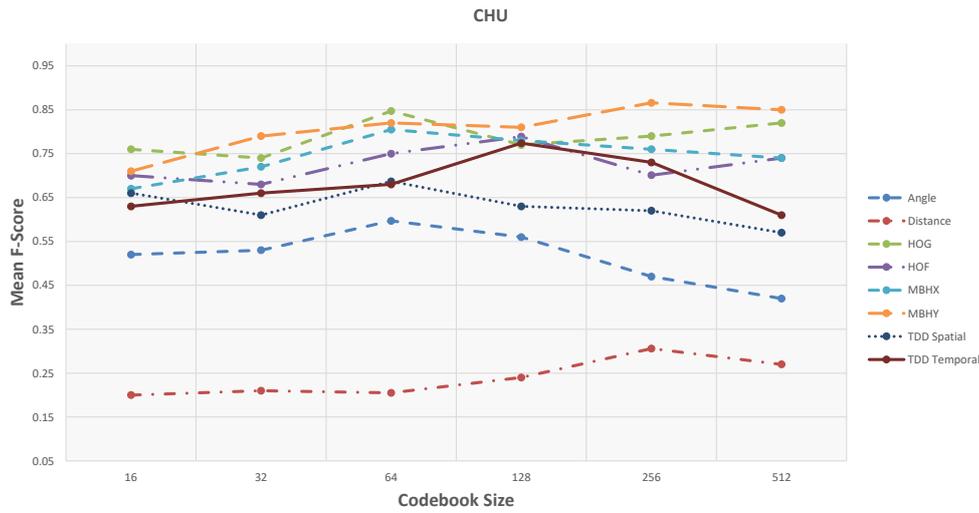
**Figure 8.** Shows F-Score values of the unsupervised framework w.r.t. codebook size on GAADR dataset.

	32			64			128			256			512		
	Prec. [%]	Rec. [%]	F-Score												
Angle	58.4	49.7	0.53	60.7	57.8	0.59	58.6	55.2	0.56	50.3	45.9	0.47	41.7	44.1	0.42
Distance	23.9	19.2	0.21	22.7	19.5	0.20	27.8	21.7	0.24	29.2	31.9	0.30	28.8	27.1	0.27
HOG	77.7	71.9	0.74	85.7	82.9	0.84	80.8	74.9	0.77	81.9	76.3	0.79	84.9	79.8	0.82
HOF	68.2	69.8	0.68	73.9	76.4	0.75	77.1	79.1	0.78	68.4	71.9	0.70	73.4	74.9	0.74
MBHX	73.4	72.1	0.72	81.3	80.4	0.80	78.6	79.2	0.78	75.2	78.3	0.76	73.4	76.2	0.74
MBHY	80.5	77.9	0.79	84.3	79.9	0.82	83.9	79.3	0.81	88.6	83.6	0.866	87.4	83.1	0.85
TDD Spatial	65.8	58.4	0.61	71.9	64.7	0.68	67.2	60.9	0.63	65.9	60.1	0.62	60.0	55.9	0.57
TDD Temporal	67.7	65.7	0.66	69.7	66.1	0.68	79.2	76.1	0.77	74.4	73.5	0.73	61.8	62.1	0.61

**Table 2.** Results regarding the unsupervised framework with different feature types on CHU dataset.

525 a big drop of performance with bigger codebook sizes. For others (HOG, HOF), medium-size codebook  
 526 performs the best. Finding an optimal codebook size is challenging. Small datasets usually work better  
 527 with smaller codebook size and as the datasets' size grows, codebook performs better. **Regardless of**  
 528 **the codebook size, MBHY descriptor performs better than other features in this dataset.** The MBH  
 529 descriptor is composed of X (MBHX) and Y (MBHY) components. As the activities involve many  
 530 vertical motions, MBHY descriptor is able to model the activities better compared to the other dense  
 531 trajectory descriptors and even deep features. It can be noticed that the performance of temporal deep  
 532 features gets better as the codebook size gets bigger. Also, motion features (TDD temporal, MBHY)  
 533 perform better than appearance features and **temporal deep features perform better than spatial TDDs.**  
 534 The reason for the lower performance of appearance features might be due to the activities performed  
 535 in a hospital environment. Hereupon, the background does not contain discriminative information  
 536 which can be encoded in activity models. It is clear that the Geometrical features perform poorly.  
 537 Daily living activities are comprised of many sub-activities with similar motion patterns related to  
 538 object interactions. It seems that geometrical features do not contain sufficient information to ensure  
 539 encoding these interactions which result in poor detection. Furthermore, the confusion matrix in figure  
 540 10 indicates that the activities with similar motion in their sub-activities are confused with each other  
 541 the most.

542 On CHU dataset, the unsupervised framework achieves promising results (Table 2 and figure  
 543 9). Similar to the GAADR dataset, the effect of codebook size is different for different descriptor  
 544 types. For MBHY descriptor, the accuracy increases as codebook size grow, whilst, it has the opposite  
 545 effect on TDD appearance features. Differently, the accuracy increases and then decreases for TDD  
 546 temporal feature. It can be observed that a bigger codebook size results in better performance. This  
 547 trend is different from GAADR dataset and the reason might be because of the larger size of this  
 548 dataset. **TDD temporal features demonstrate a better performance than deep appearance features**



**Figure 9.** Shows F-Score values of the unsupervised framework w.r.t. codebook size on CHU dataset.

549 (TDD spatial). Similarly, due to the similar background of the activities, temporal information shows  
 550 better results. MBHY achieves the best performance on this dataset. The abundance of vertical motions  
 551 in the performed activities helps the MBH descriptors to reach better recognition performance. Among  
 552 appearance features, HOG descriptor shows a better performance since it can encode the appearance  
 553 information efficiently, where it even outperforms deep appearance features. Detailed analysis (figure  
 554 10) indicates that the framework has difficulty in recognition of "Watering Plant" activity. It confuses  
 555 this activity with all the other activities. The short duration of this activity leads to insufficient capture  
 556 of local dynamic information resulting in recognition issues. The reason for the confusion of the other  
 557 activities lies mainly on similar motion patterns of the sub-activities. Moreover, this dataset consists of  
 558 activities recorded from subjects lateral view which makes recognition of those classes of activities  
 559 challenging.



**Figure 10.** Confusion matrices regarding the best configuration of the unsupervised framework on GAADR and CHU datasets (with MBHY descriptor). The values show mean accuracy (%).

560 **4.4. Comparisons**

561 This section summarizes the evaluations and comparisons conducted on GAADR 4.5, CHU 4.6  
 562 and DAHLIA 4.7 datasets.

563 The results obtained from our proposed framework on GAADR and CHU datasets are compared  
 564 with the supervised approach in [75], where videos are manually clipped. Another comparison is made

	Supervised (Manual Clipping) with HOG, Dict sz=512 [75]			Online Version of [75]			Classification by Detection SSBD [88]			Unsupervised Using Only Global Motion [86]			Hybrid [87]			Unsupervised (Proposed Method)		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Establish Account	92.2	84.3	0.88	29.1	<b>100</b>	0.45	41.67	41.67	0.41	86.2	<b>100</b>	0.92	<b>92.3</b>	<b>100</b>	0.95	86.2	<b>100</b>	0.92
Prepare Drink	92.1	<b>100</b>	0.95	69.4	<b>100</b>	0.81	80.0	96.2	0.87	<b>100</b>	78.1	0.87	<b>100</b>	92.1	0.95	<b>100</b>	100	1.0
Prepare DrugBox	94.9	85.5	<b>0.89</b>	20.2	11.7	0.14	51.28	86.96	0.64	<b>100</b>	33.34	0.50	78.5	<b>91.3</b>	0.84	<b>100</b>	33.1	0.49
Reading Article	96.2	96.2	0.96	37.8	88.6	0.52	31.88	<b>100</b>	0.48	<b>100</b>	<b>100</b>	1.0	<b>100</b>	<b>100</b>	1.0	<b>100</b>	<b>100</b>	1.0
Answer the Phone	88.5	<b>100</b>	0.93	70.1	<b>100</b>	0.82	34.29	96.0	0.50	<b>100</b>	<b>100</b>	1.0	<b>100</b>	91.2	0.95	<b>100</b>	<b>100</b>	1.0
Turn On Radio	89.4	86.7	0.88	75.1	<b>100</b>	0.85	19.86	96.55	0.32	89.0	89.0	0.89	89.1	93.4	0.91	89.1	89.3	0.89
Watering Plant	84.8	72.6	0.78	0	0	0	44.45	<b>86.36</b>	0.58	57.1	44.45	0.49	79.9	86.1	0.82	<b>100</b>	44.2	0.61
Average	91.16	89.33	0.90	43.1	71.4	0.51	43.34	86.24	0.54	90.32	77.84	0.81	91.4	<b>93.44</b>	<b>0.92</b>	<b>96.47</b>	80.94	0.84

**Table 3.** Comparison of different recognition frameworks with ours on the GAARDR dataset. The diagram shows the class-wise accuracy of each method with respect to their F-Score values. The best results in each section are indicated in bold.

	Supervised (Manual Clipping) with HOG, Dict sz=256 [75]			Online Version of [75]			Unsupervised Using Only Global Motion [86]			Hybrid			Unsupervised (Proposed Method)		
	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score	Precision [%]	Recall [%]	F-Score
Checking BusMap	<b>100</b>	97.1	<b>0.98</b>	50.1	<b>100</b>	0.66	54.54	<b>100</b>	0.70	96.1	<b>100</b>	<b>0.98</b>	80.5	86.2	0.83
Prepare DrugBox	<b>100</b>	92.3	0.95	43.2	<b>100</b>	0.60	<b>100</b>	90.1	0.94	<b>100</b>	<b>100</b>	1.0	88.2	92.7	0.90
Prepare Drink	<b>93.1</b>	<b>97.4</b>	<b>0.95</b>	38.1	76.1	0.50	80.0	84.21	0.82	88.9	96.3	0.92	<b>94.2</b>	88.5	0.91
Answer the Phone	92.2	<b>100</b>	0.95	86.7	<b>100</b>	0.92	60.1	<b>100</b>	0.75	<b>100</b>	<b>100</b>	1.0	92.4	<b>100</b>	0.96
Reading Article	97.5	94.1	0.95	36.4	92.0	0.52	<b>100</b>	81.82	0.90	<b>100</b>	<b>100</b>	1.0	93.2	87.4	0.90
Watering Plant	<b>100</b>	88.3	<b>0.93</b>	33.9	76.9	0.47	53.9	68.9	0.60	77.0	96.3	0.85	77.4	61.2	0.68
Average	<b>97.13</b>	94.87	0.95	48.06	90.83	0.61	74.75	87.50	0.78	93.66	<b>98.76</b>	<b>0.96</b>	87.65	86.00	0.86

**Table 4.** Comparison of different recognition frameworks with ours on the CHU dataset. The table below shows the detailed results of each method with respect to each class in the dataset. The best results in each section are indicated in bold.

565 with an online supervised approach that follows [75] using a sliding window scheme. The activity  
 566 models are evaluated with another version of the models [86] that does not embed local dynamic  
 567 information (in this version, the score of the local descriptor attribute is omitted and not considered  
 568 in the final score). A further comparison is performed with a Hybrid framework [87] that combines  
 569 supervised and unsupervised information in the HAM models. We additionally compare GAARDR  
 570 dataset with the produced results of another detection algorithm in [88].

#### 571 4.5. GAARDR Dataset

572 Table 3 represents the comparison of our results with the reported performance on GAARDR  
 573 dataset. In all approaches that use body motion and appearance features, the feature types with the  
 574 best performances are selected. It can be noticed that using models equipped with both global and  
 575 local motion features, the unsupervised obtains high sensitivity and precision rates. Compared to  
 576 the online version of [75], thanks to the learned zones and discovered activities, we obtain better  
 577 activity localization, thereby a better precision. Using only dense trajectories (not global motion)  
 578 this online method fails to localize activities. For the "Watering Plant" this method can not detect  
 579 any instances of this activity in the test set, hence the Precision, Recall, and F-Score rates are zero.  
 580 Compared to the unsupervised approach that either uses global motion features or body motion  
 581 features, we can see that, by combining both features, our approach achieves more discriminative and  
 582 precise models and improves both sensitivity and precision rates. Although the supervised approach  
 583 in [75] outperforms the unsupervised framework in recall and F-Score metrics, it actually does not  
 584 perform activity detection. It uses ground-truth intervals provided by manual clipping and performs  
 585 offline activity recognition which is a much simpler task. As our approach learns the scene regions, we  
 586 automatically discover the places where the activities occur, thereby we achieve precise and accurate  
 587 spatiotemporal localization with a lower cost. As scene region information is missing in the supervised  
 588 approach, it detects "Turning On Radio" while the person is inside the "Preparing Drink" region. On  
 589 this dataset, the unsupervised method always performs better than the "Online Supervised" approach  
 590 and significantly outperforms the sequential statistical boundary detection (SSBD) method. It also  
 591 outperforms another unsupervised version of the framework while no descriptor information is used  
 592 in the activity models. Only the supervised methods surpass our unsupervised models. The reason is  
 593 that the supervised method works with pre-clipped activity videos and overlooks the challenging task  
 594 of temporal segmentation of activity samples from the original video flow.

#### 4.6. CHU Dataset

Table 4 shows the results of evaluated approaches and their comparison with our results on CHU Nice Hospital dataset. In this dataset, as people tend to perform some of the activities in various regions (e.g. preparing the drink at the phone desk), it is difficult to obtain high precision rates. However, compared to the online version of the supervised method in [75], our approach detects all activities and achieves a much better precision rate. The online version of [75] again fails to detect activities accurately and misses some of the "Prepare Drink" and "Reading Article" activities and produces lots of false positives for all other activities. It cannot handle the transition states in the boundary of the activity regions (e.g. walking from telephone desk to DrugBox is detected as "Answer the Phone" activity). For this reason, a random label is assigned for transition states by the classifier, which consequently increases the rate of false positives. Compared to the Online Supervised method, we have increased the average precision rate from 48.06% to 87.65%. Compared to the unsupervised method without embedded descriptor information, we have decreased the false positive rates and increased the precision rates significantly. The highest improvements are on "Answering Phone" from 60% to 92%, "Checking BusMap" from 54.54% to 80.5%, "Prepare Drink" from 80% to 94% and "Watering Plant" from 53% to 77%. For "Reading Article" activity, there is a small increase in false positive rates, causing an incremental decrease in precision rates. This might be because of the lack of local motion information caused by staying still in a sitting posture for a long time. Since the motion representation of [86] contains only global information, it fails to distinguish activities inside the regions precisely. For instance, passing by the phone zone and answering the phone in the phone zone are considered as the same activity in their models. Hence, their unsupervised approach results in high false positive rates. In addition, we can observe that the proposed approach improves the true positive rates and increased sensitivity rates for most of the activities when it is compared to the "Only Global Motion" method.

#### 4.7. DAHLIA Dataset

Different from the two other datasets, the results on the DAHLIA dataset are compared with all the previous evaluations we could find in the literature. [89] exploits gesturelets extracted from skeleton data to compute geometrical features and detect the activities. The proposed method in [90] takes a graphical approach and poses the activity detection task as a maximum-weight connected sub-graph problem. Inspired by the Hough transformation that is successfully applied in object detection, [91] proposes a method with discriminative features to globally optimize the parameters of Hough transform and utilize it for activity segmentation in videos. Finally, our results are compared with [92] that is a supervised method with a semi-supervised component to discover sub-activities. Table 5 demonstrates our results on the DAHLIA dataset. Different metrics are used for evaluation of this dataset to enable comparison with other methods. **The table presents the best results that are produced by the generated models embedded with MBHY descriptors.** It can be noticed that in this dataset, we significantly outperform [89] and [90] in all the categories. **Efficient Linear Search (ELS)** uses geometrical features and produces poor results that are only comparable with our framework when geometrical descriptors are used in the generated models. Despite being an efficient approach, [90] demonstrates poor detection performance on Dahlia dataset. Additionally, this method only works in offline mode. [91] is another supervised method that uses both skeleton and dense trajectory descriptors and outperforms our framework only on camera view 3 while using the F-score metric. The closest performance to ours is [92] which is a supervised method and utilizes person-centered CNN features (PC-CNN) to detect sub-activities. Moreover, it has an additional post-processing step to refine the sub-activity proposals in the activity boundaries. Although our framework is totally unsupervised, we outperform this method in camera view 2 using all evaluation metrics. Similar results are obtained using different camera angles underlying the robustness of our proposed framework to viewpoint variations and different types of occlusion. This indicates that an efficient multi-view fusion method can remarkably improve the results.

	ELS [89]			Max Subgraph Search [90]			DOHT (HOG) [91]			Sub Activity [92]			Unsupervised (proposed method)		
	FA_1	F_score	IoU	FA_1	F_score	IoU	FA_1	F_score	IoU	FA_1	F_score	IoU	FA_1	F_score	IoU
<b>View 1</b>	0.18	0.18	0.11	-	0.25	0.15	0.80	0.77	0.64	<b>0.85</b>	<b>0.81</b>	<b>0.73</b>	0.84	0.79	0.70
<b>View 2</b>	0.27	0.26	0.16	-	0.18	0.10	0.81	0.79	0.66	0.87	0.82	0.75	<b>0.88</b>	<b>0.83</b>	<b>0.77</b>
<b>View 3</b>	0.52	0.55	0.39	-	0.44	0.31	0.80	<b>0.77</b>	0.65	<b>0.82</b>	0.76	<b>0.69</b>	0.79	0.73	<b>0.69</b>

**Table 5.** The activity detection results obtained on the DAHLIA. Values in bold represent the best performance.

644 In overall, although our unsupervised framework does not utilize any supervised information, it  
645 achieved promising recognition performances. Compared to the fully supervised hybrid method [87],  
646 the unsupervised framework obtains acceptable and competitive results in the detection of most of  
647 the activities. However, the high performance of the hybrid method comes with the cost of human  
648 supervision. In the hybrid method, a supervised **Support Vector Machine (SVM)** classifier is trained  
649 with the ground-truth annotation provided by a human. The main benefits of the unsupervised  
650 method are automatic online clipping and detection of activities as well as unsupervised modeling and  
651 recognition. With all these benefits, the marginal difference in the recognition rate of the unsupervised  
652 method relative to supervised counterparts is admissible.

## 653 5. Conclusions

654 An online unsupervised framework is proposed for detection of daily living activities, particularly  
655 for elderly monitoring. To create the activity models, we benefited from the superiority of unsupervised  
656 approaches on representing global motion patterns. Then, discriminative local motion features were  
657 employed in order to generate a more accurate model of activity dynamics. Thanks to the proposed  
658 scene model, online recognition of activities can be performed with reduced user interaction for  
659 clipping and labeling a huge amount of short-term actions which are essential for most of the previously  
660 proposed methods. Our extensive evaluations on three datasets revealed that our proposed framework  
661 is capable of detecting and recognizing activities in challenging scenarios. The evaluations were  
662 intentionally conducted on the datasets recorded in nursing homes, hospitals and smart homes to  
663 examine the implication of the method on ambient surveillance in such environments. Further work  
664 will investigate how to generate generic models that can detect activities in any environment with  
665 minimum modification of the models. Our goal is to use the developed framework in the evaluation of  
666 long-term video recordings in nursing homes and to assess the performance of the subjects to impose  
667 early interventions which will result in early diagnosis of cognitive disorders, especially Alzheimer's  
668 disease.

669 **Conflicts of Interest:** "The authors declare no conflict of interest."

## 670 Abbreviations

671 The following abbreviations are used in this manuscript:

672

ADL	Activities of Daily Living
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
C3D	Convolution3D
TCN	Temporal Convolutional Network
HDP	Hierarchical Dirichlet Process
HOG	Histogram of Oriented Gradients
HOF	Histogram of Optical Flow
MBH	Motion Boundaries Histogram
MBHX	Motion Boundaries Histogram in X axis
MBHY	Motion Boundaries Histogram in Y axis
TSD	Trajectory Shape Descriptor
TDD	Trajectory-Pooled Deep-Convolutional Descriptors
BIC	Bayesian Information Criterion
SR	Scene Region
PE	Primitive Event
DA	Discovered Activity
FV	Fisher Vector
HAM	Hierarchical Activity Model
MAP	Maximum A Posteriori
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
TPR	True Positive Rate
PPV	Positive Predictive Value
IoU	Intersection over Union
SSBD	Sequential statistical boundary detection
ELS	Efficient Linear Search
PC-CNN	Person-Centered CNN
SVM	Support Vector Machine

673

674 **References**

- 675 1. Heilbron, F.C.; Barrios, W.; Escorcia, V.; Ghanem, B. Scc: Semantic context cascade for efficient action  
676 detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017, pp.  
677 3175–3184.
- 678 2. Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; Lin, D. Temporal action detection with structured segment  
679 networks. Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2914–2923.
- 680 3. Xu, H.; Das, A.; Saenko, K. R-c3d: Region convolutional 3d network for temporal activity detection.  
681 Proceedings of the IEEE international conference on computer vision, 2017, pp. 5783–5792.
- 682 4. Shou, Z.; Wang, D.; Chang, S.F. Temporal action localization in untrimmed videos via multi-stage cnns.  
683 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1049–1058.
- 684 5. Oneata, D.; Verbeek, J.; Schmid, C. The lear submission at thumos 2014 **2013**.
- 685 6. Wang, L.; Qiao, Y.; Tang, X. Action recognition and detection by combining motion and appearance  
686 features. *THUMOS14 Action Recognition Challenge* **2014**, 1, 2.
- 687 7. Wang, L.; Qiao, Y.; Tang, X.; Van Gool, L. Actionness estimation using hybrid fully convolutional networks.  
688 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2708–2717.
- 689 8. Caba Heilbron, F.; Carlos Niebles, J.; Ghanem, B. Fast temporal activity proposals for efficient detection of  
690 human actions in untrimmed videos. Proceedings of the IEEE conference on computer vision and pattern  
691 recognition, 2016, pp. 1914–1923.
- 692 9. Escorcia, V.; Heilbron, F.C.; Niebles, J.C.; Ghanem, B. Daps: Deep action proposals for action understanding.  
693 European Conference on Computer Vision. Springer, 2016, pp. 768–784.

- 694 10. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.;  
695 Bernstein, M.; others. Imagenet large scale visual recognition challenge. *International journal of computer*  
696 *vision* **2015**, *115*, 211–252.
- 697 11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal  
698 networks. *Advances in neural information processing systems*, 2015, pp. 91–99.
- 699 12. Hosang, J.; Benenson, R.; Dollár, P.; Schiele, B. What makes for effective detection proposals? *IEEE*  
700 *transactions on pattern analysis and machine intelligence* **2016**, *38*, 814–830.
- 701 13. Marszałek, M.; Laptev, I.; Schmid, C. Actions in context. *CVPR 2009-IEEE Conference on Computer Vision*  
702 *& Pattern Recognition*. IEEE Computer Society, 2009, pp. 2929–2936.
- 703 14. Wu, Z.; Fu, Y.; Jiang, Y.G.; Sigal, L. Harnessing object and scene semantics for large-scale video  
704 understanding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*,  
705 2016, pp. 3112–3121.
- 706 15. Jain, M.; Van Gemert, J.C.; Snoek, C.G. What do 15,000 object categories tell us about classifying and  
707 localizing actions? *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015,  
708 pp. 46–55.
- 709 16. Wang, H.; Schmid, C. Action recognition with improved trajectories. *Proceedings of the IEEE International*  
710 *Conference on Computer Vision*, 2013, pp. 3551–3558.
- 711 17. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d  
712 convolutional networks. *Proceedings of the IEEE international conference on computer vision*, 2015, pp.  
713 4489–4497.
- 714 18. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv*  
715 *preprint arXiv:1409.1556* **2014**.
- 716 19. Sadanand, S.; Corso, J.J. Action bank: A high-level representation of activity in video. *2012 IEEE Conference*  
717 *on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1234–1241.
- 718 20. Liu, J.; Kuipers, B.; Savarese, S. Recognizing human actions by attributes. *CVPR 2011*. IEEE, 2011, pp.  
719 3337–3344.
- 720 21. Bojanowski, P.; Lajugie, R.; Bach, F.; Laptev, I.; Ponce, J.; Schmid, C.; Sivic, J. Weakly supervised action  
721 labeling in videos under ordering constraints. *European Conference on Computer Vision*. Springer, 2014,  
722 pp. 628–643.
- 723 22. Duchenne, O.; Laptev, I.; Sivic, J.; Bach, F.R.; Ponce, J. Automatic annotation of human actions in video.  
724 *ICCV*, 2009, Vol. 1, pp. 3–2.
- 725 23. Tian, Y.; Sukthankar, R.; Shah, M. Spatiotemporal deformable part models for action detection. *Proceedings*  
726 *of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2642–2649.
- 727 24. Ni, B.; Paramathayalan, V.R.; Moulin, P. Multiple granularity analysis for fine-grained action detection.  
728 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 756–763.
- 729 25. Bhattacharya, S.; Kalayeh, M.M.; Sukthankar, R.; Shah, M. Recognition of complex events: Exploiting  
730 temporal dynamics between underlying concepts. *Proceedings of the IEEE conference on computer vision*  
731 *and pattern recognition*, 2014, pp. 2235–2242.
- 732 26. Tang, K.; Fei-Fei, L.; Koller, D. Learning latent temporal structure for complex event detection. *2012 IEEE*  
733 *Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1250–1257.
- 734 27. Vo, N.N.; Bobick, A.F. From stochastic grammar to bayes network: Probabilistic parsing of complex activity.  
735 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2641–2648.
- 736 28. Wang, X.; Ji, Q. A hierarchical context model for event recognition in surveillance video. *Proceedings of*  
737 *the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2561–2568.
- 738 29. Modiri Assari, S.; Roshan Zamir, A.; Shah, M. Video classification using semantic concept co-occurrences.  
739 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2529–2536.
- 740 30. Niebles, J.C.; Chen, C.W.; Fei-Fei, L. Modeling temporal structure of decomposable motion segments for  
741 activity classification. *European conference on computer vision*. Springer, 2010, pp. 392–405.
- 742 31. Koppula, H.; Saxena, A. Learning spatio-temporal structure from rgb-d videos for human activity detection  
743 and anticipation. *International conference on machine learning*, 2013, pp. 792–800.
- 744 32. Jones, S.; Shao, L. Unsupervised spectral dual assignment clustering of human actions in context.  
745 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 604–611.

- 746 33. Yang, Y.; Saleemi, I.; Shah, M. Discovering motion primitives for unsupervised grouping and one-shot  
747 learning of human actions, gestures, and expressions. *IEEE transactions on pattern analysis and machine*  
748 *intelligence* **2013**, *35*, 1635–1648.
- 749 34. Morris, B.; Trivedi, M. Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and  
750 Long-Term Adaptive Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2011**,  
751 *33*, 2287–2301. doi:10.1109/TPAMI.2011.64.
- 752 35. Gao, Q.; Sun, S. Trajectory-based human activity recognition with hierarchical Dirichlet process hidden  
753 Markov models. Proceedings of the 1st IEEE China Summit and International Conference on Signal and  
754 Information Processing, 2013.
- 755 36. Hu, W.; Xiao, X.; Fu, Z.; Xie, D.; Tan, T.; Maybank, S. A system for learning statistical motion patterns.  
756 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2006**, *28*, 1450–1464.
- 757 37. Wang, L.; Qiao, Y.; Tang, X. Action recognition with trajectory-pooled deep-convolutional descriptors.  
758 Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4305–4314.
- 759 38. Mathe, S.; Sminchisescu, C. Actions in the eye: Dynamic gaze datasets and learnt saliency models for  
760 visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *37*, 1408–1424.
- 761 39. Hoai, M.; Lan, Z.Z.; De la Torre, F. Joint segmentation and classification of human actions in video. CVPR  
762 2011. IEEE, 2011, pp. 3265–3272.
- 763 40. Shi, Q.; Cheng, L.; Wang, L.; Smola, A. Human action segmentation and recognition using discriminative  
764 semi-markov models. *International journal of computer vision* **2011**, *93*, 22–32.
- 765 41. Kuehne, H.; Arslan, A.; Serre, T. The language of actions: Recovering the syntax and semantics of  
766 goal-directed human activities. Proceedings of the IEEE conference on computer vision and pattern  
767 recognition, 2014, pp. 780–787.
- 768 42. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification  
769 with convolutional neural networks. Proceedings of the IEEE conference on Computer Vision and Pattern  
770 Recognition, 2014, pp. 1725–1732.
- 771 43. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos.  
772 Advances in neural information processing systems, 2014, pp. 568–576.
- 773 44. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T.  
774 Long-term recurrent convolutional networks for visual recognition and description. Proceedings of the  
775 IEEE conference on computer vision and pattern recognition, 2015, pp. 2625–2634.
- 776 45. Yue-Hei Ng, J.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; Toderici, G. Beyond short  
777 snippets: Deep networks for video classification. Proceedings of the IEEE conference on computer vision  
778 and pattern recognition, 2015, pp. 4694–4702.
- 779 46. Karaman, S.; Seidenari, L.; Del Bimbo, A. Fast saliency based pooling of fisher encoded dense trajectories.  
780 ECCV THUMOS Workshop, 2014, Vol. 1, p. 5.
- 781 47. Gaidon, A.; Harchaoui, Z.; Schmid, C. Temporal localization of actions with actoms. *IEEE transactions on*  
782 *pattern analysis and machine intelligence* **2013**, *35*, 2782–2795.
- 783 48. Tang, K.; Yao, B.; Fei-Fei, L.; Koller, D. Combining the right features for complex event recognition.  
784 Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2696–2703.
- 785 49. De Geest, R.; Gavves, E.; Ghodrati, A.; Li, Z.; Snoek, C.; Tuytelaars, T. Online action detection. European  
786 Conference on Computer Vision. Springer, 2016, pp. 269–284.
- 787 50. Yeung, S.; Russakovsky, O.; Mori, G.; Fei-Fei, L. End-to-end learning of action detection from frame  
788 glimpses in videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,  
789 2016, pp. 2678–2687.
- 790 51. Montes, A.; Salvador, A.; Pascual, S.; Giro-i Nieto, X. Temporal activity detection in untrimmed videos  
791 with recurrent neural networks. *arXiv preprint arXiv:1608.08128* **2016**.
- 792 52. Ma, S.; Sigal, L.; Sclaroff, S. Learning activity progression in lstms for activity detection and early detection.  
793 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1942–1950.
- 794 53. Oord, A.v.d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.;  
795 Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* **2016**.
- 796 54. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks for action  
797 segmentation and detection. proceedings of the IEEE Conference on Computer Vision and Pattern  
798 Recognition, 2017, pp. 156–165.

- 799 55. Chen, W.; Xiong, C.; Xu, R.; Corso, J.J. Actionness ranking with lattice conditional ordinal random fields.  
800 Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 748–755.
- 801 56. Qiu, H.; Zheng, Y.; Ye, H.; Lu, Y.; Wang, F.; He, L. Precise temporal action localization by evolving temporal  
802 proposals. Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval. ACM,  
803 2018, pp. 388–396.
- 804 57. Gkioxari, G.; Malik, J. Finding action tubes. Proceedings of the IEEE conference on computer vision and  
805 pattern recognition, 2015, pp. 759–768.
- 806 58. Mettes, P.; Van Gemert, J.C.; Snoek, C.G. Spot on: Action localization from pointly-supervised proposals.  
807 European conference on computer vision. Springer, 2016, pp. 437–453.
- 808 59. Weinzaepfel, P.; Harchaoui, Z.; Schmid, C. Learning to track for spatio-temporal action localization.  
809 Proceedings of the IEEE international conference on computer vision, 2015, pp. 3164–3172.
- 810 60. Jiang, Z.; Lin, Z.; Davis, L.S. A unified tree-based framework for joint action localization, recognition and  
811 segmentation. *Computer Vision and Image Understanding* **2013**, *117*, 1345–1355.
- 812 61. Soomro, K.; Idrees, H.; Shah, M. Action localization in videos through context walk. Proceedings of the  
813 IEEE international conference on computer vision, 2015, pp. 3280–3288.
- 814 62. Jain, M.; Van Gemert, J.; Jégou, H.; Bouthemy, P.; Snoek, C.G. Action localization with tubelets from motion.  
815 Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 740–747.
- 816 63. Yu, G.; Yuan, J. Fast action proposals for human action detection and search. Proceedings of the IEEE  
817 conference on computer vision and pattern recognition, 2015, pp. 1302–1311.
- 818 64. Guerra Filho, G.; Aloimonos, Y. A language for human action. *Computer* **2007**, *40*, 42–51.
- 819 65. Fox, E.B.; Hughes, M.C.; Sudderth, E.B.; Jordan, M.I.; others. Joint modeling of multiple time series via  
820 the beta process with application to motion capture segmentation. *The Annals of Applied Statistics* **2014**,  
821 *8*, 1281–1313.
- 822 66. Emonet, R.; Varadarajan, J.; Odobez, J.M. Temporal analysis of motif mixtures using dirichlet processes.  
823 *IEEE transactions on pattern analysis and machine intelligence* **2013**, *36*, 140–156.
- 824 67. Brattoli, B.; Buchler, U.; Wahl, A.S.; Schwab, M.E.; Ommer, B. Lstm self-supervision for detailed behavior  
825 analysis. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp.  
826 6466–6475.
- 827 68. Wang, X.; Gupta, A. Unsupervised learning of visual representations using videos. Proceedings of the  
828 IEEE International Conference on Computer Vision, 2015, pp. 2794–2802.
- 829 69. Cherian, A.; Fernando, B.; Harandi, M.; Gould, S. Generalized rank pooling for activity recognition.  
830 Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3222–3231.
- 831 70. Fernando, B.; Gavves, E.; Oramas, J.M.; Ghodrati, A.; Tuytelaars, T. Modeling video evolution for action  
832 recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp.  
833 5378–5387.
- 834 71. Lee, H.Y.; Huang, J.B.; Singh, M.; Yang, M.H. Unsupervised representation learning by sorting sequences.  
835 Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 667–676.
- 836 72. Ramanathan, V.; Tang, K.; Mori, G.; Fei-Fei, L. Learning temporal embeddings for complex video analysis.  
837 Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4471–4479.
- 838 73. Milbich, T.; Bautista, M.; Sutter, E.; Ommer, B. Unsupervised video understanding by reconciliation of  
839 posture similarities. Proceedings of the IEEE International Conference on Computer Vision, 2017, pp.  
840 4394–4404.
- 841 74. Crispim-Junior, C.; Gómez Uría, A.; Strumia, C.; Koperski, M.; König, A.; Negin, F.; Cosar, S.; Nghiem, A.;  
842 Chau, D.; Charpiat, G.; others. Online recognition of daily activities by color-depth sensing and knowledge  
843 models. *Sensors* **2017**, *17*, 1528.
- 844 75. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Action Recognition by Dense Trajectories. IEEE Conference on  
845 Computer Vision & Pattern Recognition, ; 2011; pp. 3169–3176.
- 846 76. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. IEEE Computer Society  
847 Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005., 2005, Vol. 1, pp. 886–893 vol.  
848 1. doi:10.1109/CVPR.2005.177.
- 849 77. Agahian, S.; Negin, F.; Köse, C. Improving bag-of-poses with semi-temporal pose descriptors for  
850 skeleton-based action recognition. *The Visual Computer* **2019**, *35*, 591–607.

- 851 78. Nghiem, A.T.; Auvinet, E.; Meunier, J. Head detection using Kinect camera and its application to fall  
852 detection. *ISSPA*, 2012, pp. 164–169.
- 853 79. Anh, N.T.L.; Khan, F.M.; Negin, F.; Bremond, F. Multi-object tracking using multi-channel part appearance  
854 representation. 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance  
855 (AVSS). IEEE, 2017, pp. 1–6.
- 856 80. Kuhn, H.W. The Hungarian method for the assignment problem. *Naval research logistics quarterly* **1955**,  
857 *2*, 83–97.
- 858 81. Chau, D.P.; Thonnat, M.; Bremond, F. Automatic parameter adaptation for multi-object tracking.  
859 International Conference on Computer Vision Systems. Springer, 2013, pp. 244–253.
- 860 82. Pelleg, D.; Moore, A.W.; others. X-means: Extending k-means with efficient estimation of the number of  
861 clusters. *Icml*, 2000, Vol. 1, pp. 727–734.
- 862 83. Sánchez, J.; Perronnin, F.; Mensink, T.; Verbeek, J. Image classification with the fisher vector: Theory and  
863 practice. *International journal of computer vision* **2013**, *105*, 222–245.
- 864 84. Karakostas, A.; Briassouli, A.; Avgerinakis, K.; Kompatsiaris, I.; M., T. The Dem@Care Experiments and  
865 Datasets: a Technical Report. Technical report, 2014.
- 866 85. Vaquette, G.; Orcesi, A.; Lucat, L.; Achard, C. The DAily Home LIfe Activity Dataset: A High Semantic  
867 Activity Dataset for Online Recognition. *FG 2017*, May. doi:10.1109/FG.2017.67.
- 868 86. Negin, F.; Cogar, S.; Bremond, F.; Koperski, M. Generating unsupervised models for online long-term daily  
869 living activity recognition. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). IEEE, 2015,  
870 pp. 186–190.
- 871 87. Negin, F.; Koperski, M.; Crispim, C.F.; Bremond, F.; Coşar, S.; Avgerinakis, K. A hybrid framework  
872 for online recognition of activities of daily living in real-world settings. 2016 13th IEEE International  
873 Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2016, pp. 37–43.
- 874 88. Avgerinakis, K.; Briassouli, A.; Kompatsiaris, I. Activity detection using sequential statistical boundary  
875 detection (ssbd). to appear in *Computer Vision and Image Understanding*. CVIU, 2015.
- 876 89. Meshry, M.; Hussein, M.E.; Toriki, M. Linear-time online action detection from 3D skeletal data using bags  
877 of gesturelets. *WACV 2016*.
- 878 90. Chen, C.; Grauman, K. Efficient Activity Detection in Untrimmed Video with Max-Subgraph Search. *IEEE*  
879 *Trans. Pattern Anal. Mach. Intell.* **2017**.
- 880 91. Chan-Hon-Tong, A.; Achard, C.; Lucat, L. Deeply Optimized Hough Transform: Application to Action  
881 Segmentation. *ICIAP 2013*.
- 882 92. Negin, F.; Goel, A.; Abubakr, A.G.; Bremond, F.; Francesca, G. Online detection of long-term daily living  
883 activities by weakly supervised recognition of sub-activities. 2018 15th IEEE International Conference on  
884 Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2018, pp. 1–6.